

# Deep learning in color: towards automated quark/gluon jet discrimination

---

Patrick T. Komiske,<sup>a</sup> Eric M. Metodiev,<sup>a</sup> and Matthew D. Schwartz<sup>b</sup>

<sup>a</sup>*Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA*

<sup>b</sup>*Department of Physics, Harvard University, Cambridge, MA 02138, USA*

*E-mail:* [pkomiske@mit.edu](mailto:pkomiske@mit.edu), [metodiev@mit.edu](mailto:metodiev@mit.edu),  
[schwartz@physics.harvard.edu](mailto:schwartz@physics.harvard.edu)

**ABSTRACT:** Artificial intelligence offers the potential to automate challenging data-processing tasks in collider physics. To establish its prospects, we explore to what extent deep learning with convolutional neural networks can discriminate quark and gluon jets better than observables designed by physicists. Our approach builds upon the paradigm that a jet can be treated as an image, with intensity given by the local calorimeter deposits. We supplement this construction by adding color to the images, with red, green and blue intensities given by the transverse momentum in charged particles, transverse momentum in neutral particles, and pixel-level charged particle counts. Overall, the deep networks match or outperform traditional jet variables. We also find that, while various simulations produce different quark and gluon jets, the neural networks are surprisingly insensitive to these differences, similar to traditional observables. This suggests that the networks can extract robust physical information from imperfect simulations.

---

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Deep Neural Networks</b>	<b>3</b>
<b>3</b>	<b>Event Generation and Network Architecture</b>	<b>5</b>
3.1	Pre-processing	6
3.2	Network architecture	7
3.3	Jet images in color	9
<b>4</b>	<b>Fisher jets and a look inside the networks</b>	<b>10</b>
<b>5</b>	<b>Network performance</b>	<b>12</b>
5.1	Colored Jet Images	13
5.2	Merge layers	14
5.3	Image-size dependance	15
5.4	Event generator dependance	15
<b>6</b>	<b>Conclusions</b>	<b>17</b>

---

## 1 Introduction

High energy particle collisions produce an enormous amount of data. For example, the Large Hadron Collider (LHC) is currently generating petabytes per year. Sorting through all of this data is a herculean task, but one that should be amenable to processing using modern developments in data science and artificial intelligence. Neural networks and other approaches already play a significant role in LHC data processing, particularly at the lowest levels, in the electronics [1], or in intermediate level tasks such as tagging of bottom quarks [2, 3] or tau identification [4], and in matching data to non-perturbative physics, such as in fitting parton distribution functions [5]. They have also been used effectively for distinguishing certain signals from their known backgrounds [6–9]. For these applications, one generally constructs a set of physically-motivated but often highly-correlated observables, such as the dijet mass, or angular distributions of decay products, and the neural network is used to combine them into a single discriminant. One might imagine, however, that such an approach is limited by the creativity of physicists who construct the input observables to begin with. Thus, it is important to determine how well neural networks can do at discriminating two event samples with minimal physical input. In particular, in this paper we explore whether artificial

intelligence can do well at the challenging task of distinguishing quark jets from gluon jets using data in reasonably raw form rather than using carefully constructed observables.

An arguably minimal approach to processing the LHC data is the “jet images” approach introduced in [10, 11]. The idea behind jet images is to treat the energy deposits in a calorimeter as intensities in a 2D image. Then one can apply sophisticated algorithms developed for image recognition to particle physics. This and related neural network approaches were used for boosted  $W$  boson tagging in [10] and [12], top tagging in [11], heavy-flavor tagging in [13, 14], and comparing parton shower uncertainties in [15]. In many of these studies the data was manipulated using some physical insight before being sent into the network. For example, boosted hadronically-decaying  $W$  bosons generally look like large “fat jets” with two fairly well-defined subjets. Using this insight, in [10], each jet image was rotated to align with a moment of the fat jet. While some pre-processing is always useful to make the network training more efficient, we will attempt to avoid any pre-processing motivated by physical insight into the samples. For example, we allow generic pre-processing, like normalizing the pixel intensities [16], but avoid steps like looking for subjets that we expect in signal samples but not in background samples.

One of the most challenging tasks in collider physics is to tell apart jets initiated by light quarks from those initiated by gluons. This problem has been studied for decades [17–20] with a fair amount of recent activity [21–23]. Gluon acceptance efficiencies from 20% to 5% are achievable at a 50% quark acceptance working point [24]. The large variation is a result of assumptions about detector properties (better angular resolution produces better discrimination) and which simulations are used to approximate the data (e.g. quark and gluon jets are more distinguishable in Pythia than in Herwig). Some general lessons from [24] and [25] were that there are essentially two complementary types of observables: shape and count observables. Shape observables are quantities such as the width or girth of a jet, its mass, or an energy-correlation function; count observables are quantities such as the number of charged particles in a jet or the number of distinct calorimeter cells that are triggered. A recent study [26] explored the discrepancy among the simulations, finding that programs with more sophisticated parton showers, such as Vincia, tend to perform intermediately between Herwig and Pythia. In studies with actual data [27], the relevant observables also seemed to fall somewhere in between Herwig and Pythia, suggesting that the improved parton shower models may produce more accurate simulations.

From these studies, one may draw a couple of critical observations. First, current simulations of quark and gluon jet properties are not completely trustworthy. This naturally suggests that one should use discriminants with a solid theoretical justification, so that one does not have to rely on the simulations. The set of observables constructed in [24, 25], such as girth and track count, were all motivated physically and so one expects them to work on data whether or not the simulations agree with the data. A typical semi-classical theory argument is that gluon jets should have about twice as much radiation as quark jets ( $C_A/C_F = 9/4$ ), making them wider and with more particles. This Casimir scaling only goes so far, however, as it does not tell us if track count and girth should be complementary or not. Moreover,

detector effects and hadronization have an important effect on the jet substructure that is difficult to approach analytically.

A second observation is that the theory and simulations seem to be improving. For example, it is already possible to make trustworthy calculations beyond the semi-classical limit (e.g. in [28] it was shown that soft-drop allows for an unambiguous quark/gluon jet definition). First-principles calculations of correlations among observables are also being explored [22]. Thus, it is easy to imagine that the simulations will be trustworthy before long. Therefore, our approach in this paper will be to pretend that we live in the future, where the simulations are in fantastic agreement with data. In such an ideal world, are physically motivated observables still necessary, or can artificial intelligence, through deep neural networks, truly find an optimal solution to the quark/gluon discrimination problem?

This paper is intended to be readable by an audience with minimal previous exposure to deep learning. We begin in Section 2 with an introduction to some of the terminology used in the neural network community and an overview of some of the insights from recent years that have led to the rapid development of this technology. In Section 3 we discuss our data simulation and network architecture, including our innovation of adding multiple channels (“colors”) to the jet images. Section 4 explores the fisher-jet approach, following [10] and its connection to convolutional filters. The network performance is discussed in Section 5 and our conclusions are presented in Section 6.

## 2 Deep Neural Networks

Artificial neural networks (ANNs or NNs) are a powerful tool in machine learning and have been successfully applied to many problems in fields such as computer vision [16], natural language processing [29], and physics [30, 31]. Recent comprehensive introductions to neural networks and deep learning can be found in [32] and [33].

The basic goal of a neural network is to learn a function from a set of fixed-size inputs to a fixed-size output. The network consists of the *input layer*, one or more *intermediate* or *hidden* layers consisting of a set of *units* or *nodes*, and an *output layer*. In a *feed-forward* neural network, the layers are ordered and each unit of a layer connects to some subset of the units of previous layers. A layer is *dense* if each of its units connects to all of the units in the previous layer. A network is *deep* if there are several hidden layers. Each connection between units in adjacent layers has a *weight* and each unit has a *bias*.

The value of a unit in a non-input layer is obtained by multiplying the values of its inputs by the respective weights of their connections, summing these values, and adding the bias. This sum is then operated on by an *activation function*. The idea of an activation function is inspired by biological neurons that fire after a certain threshold is reached. Correspondingly, these functions were traditionally taken to be smoothed-out step functions, like a sigmoid or logistic function. One of the insights which allowed the rapid progress in deep learning in recent years is the observation that training can be easier with different activation functions.

For example, the sigmoid has a nearly-vanishing gradient for inputs far from zero, which can lead to *saturation*, whereby the network becomes insensitive to changes in input unit values. To avoid this, modern applications in computer vision typically use the rectified linear unit (ReLU) [34], with  $\text{ReLU}(x) = \max\{0, x\}$ , for an activation function. ReLUs are computationally fast to evaluate and avoid saturation since their derivative is 1 for any positive value of the input.

To learn a function with a neural network, the weights and biases are typically determined through *supervised learning*, whereby the network is shown many examples of the input for which the true value of the function is known. In the case of classification, the network is shown examples for which the true class is known. A *loss function* encapsulates the difference between the network output and the true class. Minimization of the loss function proceeds by calculating the gradient of the loss function with respect to the weights and biases of the network using the *back-propagation algorithm*, and updating these parameters via stochastic gradient descent. In this way, the network is trained to classify new examples.

It can be proven that even a network with as few as one hidden layer (a shallow network) can approximate any well-behaved function to arbitrary accuracy if it has sufficiently many units [35]. Deep networks offer the potential to approach this optimum much more efficiently, by having more layers with fewer units, than dense, shallow networks [36, 37]. The additional hidden layers allow the network to identify low level features in the early layers and more abstract, higher-level features in the later layers.

Until recently, complications such as long training times for large datasets, unit saturation, and overfitting have prevented the effective usage of deep networks. The increasing availability of computing power, especially specialized graphics processing units (GPUs), has sped up training times of deep networks. The introduction of new activation functions such as the ReLU have alleviated saturation issues by avoiding the vanishing gradient problem. Another problem with in machine learning is *overfitting*, where a model picks up on overly-fine details of the training samples and performs poorly on test samples. The term *regularization* refers to methods introduced to avoid overfitting. An effective regularization method for neural networks is *dropout* [38], in which a random subset of units are ignored during each training update in order to avoid over-dependence on any particular part of the network. Dropout works well to regularize large and complex networks.

Another important development has been the adoption of convolutional neural networks (CNNs), namely networks which include one or more *convolutional layers*, as a standard tool in image recognition problems. A convolutional layer is computed from the previous layer by convolving with a filter. A *filter* is an  $n \times n$  grid of weights. The convolution multiplies this filter by a patch of the previous layer, sums the values, and adds a bias, and then applies the activation function. The filter is shifted along the image by a *stride length*, usually taken to be 1 unit. For instance starting with a  $20 \times 20$  pixel image, a  $5 \times 5$  filter can be placed at  $16 \times 16$  locations on the image, so with a stride length of 1 the convolutional layer has  $16 \times 16$  units. Note that the same filter is used for each offset, so only the  $5 \times 5$  weights in the filter and the bias are independently trained. Typically many different filters connected

to the same inputs are trained, starting from different initial values (in our application, each convolutional layer will have 64 filters). Different filters can pick up on different and often complementary aspects of the image.

Finally, CNNs typically have *max-pooling* layers. These layers are simply down-samplings, where each unit takes the the maximum value in  $m \times m$  patches in a convolutional layer. Maxpooling reduces the number of parameters, which effectively allows the network to focus on relevant features. For instance, a  $16 \times 16$  unit convolutional layer would be downsampled by  $2 \times 2$  maxpooling to an  $8 \times 8$  unit layer. This reduced layer is then taken as input into the next layer of the network.

### 3 Event Generation and Network Architecture

Previous studies of quark and gluon jet discrimination have found a notable difference among different simulations [24, 26]. Thus we generated events using both Pythia 8.219 [39] and Herwig 7.0 [40, 41] using their respective default tunings and shower parameters. We simulated dijet events in  $pp$  collisions at  $\sqrt{s} = 13$  TeV. The light-quark ( $u, d, s$ ) initiated jets come from parton-level hard processes  $qq \rightarrow qq$ ,  $q\bar{q} \rightarrow q\bar{q}$  or  $gg \rightarrow qq$ ; the gluon-initiated jets come from  $gg \rightarrow gg$  or  $q\bar{q} \rightarrow gg$  for gluon production. We turned off  $qg \rightarrow qg$  for simplicity. Final state particles with pseudorapidity  $|\eta| < 2.5$  were kept and neutrinos were discarded. The resulting events were then clustered with FastJet [42] using the anti- $k_T$  algorithm [43] with a jet radius of  $R = 0.4$ .

Four jet-transverse-momentum ( $p_T$ ) ranges were used: 100–110 GeV, 200–220 GeV, 500–550 GeV, 1000–1100 GeV. The parton-level  $p_T$  cuts were chosen 20% broader than these ranges in order to ensure that jet  $p_T$  distributions were not distorted from distributions with no parton-level cuts. For each  $p_T$  range, 100k each of quark/gluon Pythia jets were generated. For the 200–220 GeV and 1000–1100 GeV ranges, 100k each of quark/gluon Herwig jets were generated for a comparison of different MCs. The four  $p_T$  ranges will be referred to by their lower limits: 100 GeV, 200 GeV, 500 GeV, 1000 GeV. In each set of 100k events, 90k were used for training and 10k for testing.

For each jet in a given  $p_T$  range, we constructed a jet image, following [10, 12]. The images are square arrays in  $(\eta, \phi)$ -space with each pixel given by the total  $p_T$  deposited in the associated region. The images are used as the input layer to the neural network. It is helpful to have a center pixel, so we chose odd numbers of grid lengths, e.g.  $33 \times 33$  pixels. For a jet radius  $R$ , the image has size  $2R \times 2R$ . In this paper we used  $R = 0.4$ , which for  $33 \times 33$  pixels corresponds to a  $\Delta\eta = \Delta\phi = 0.024$ . The discretization into  $\Delta\eta \times \Delta\phi = 0.024 \times 0.024$  pixels also acts as a kind of coarse-graining often used as a primitive detector simulator (e.g. see [44]). We follow [12] in using transverse momentum as our pixel intensity rather than the energy due to the invariance of the transverse momentum under longitudinal boosts.

The approach taken in this paper regards the input values as exact, neglecting issues related to measurement uncertainty. Unlike in typical computer vision applications, the inputs to neural networks in physics applications have uncertainties associated with them.

Neural networks are capable of sensible error propagation when supplied with input values with errors. One approach taken by the NNPDF collaboration [45] is to sample several “replicated” datasets from the distribution determined by the measured data with errors and to train a separate network on each of them. Then the uncertainty of a prediction on a new data point can be determined from the distribution of model outputs on that data point.

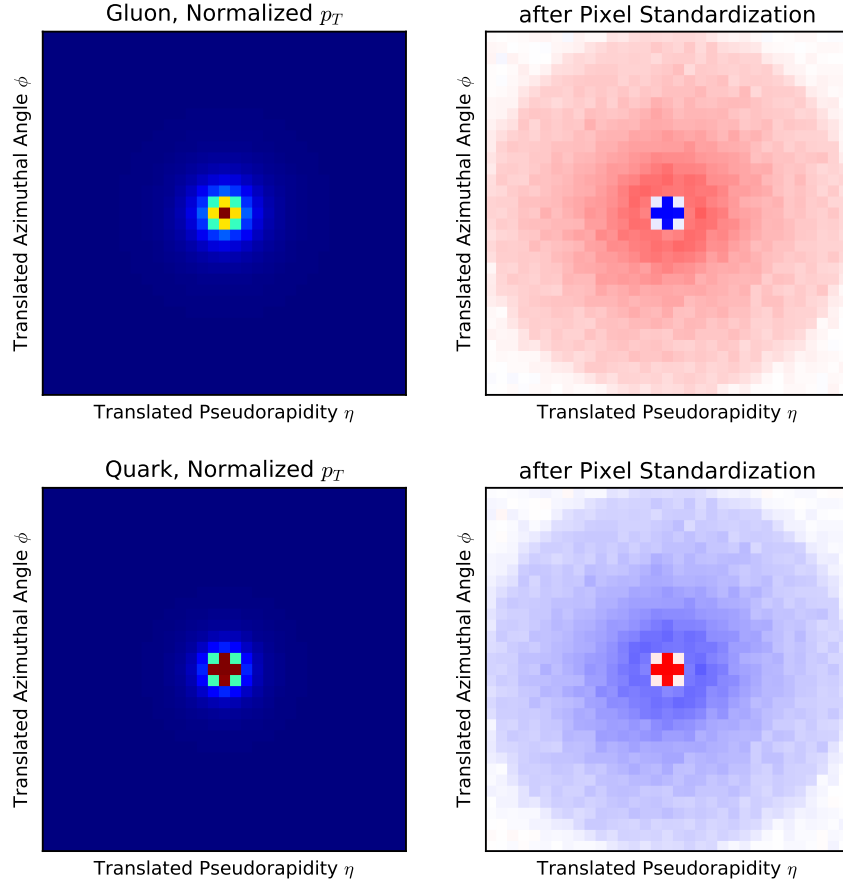
### 3.1 Pre-processing

The following series of data-motivated pre-processing steps were applied to the jet images:

1. **Center:** Center the jet image by translating in  $(\eta, \phi)$  so that the total  $p_T$ -weighted centroid pixel is at  $(\eta, \phi) = (0, 0)$ . This operation corresponds to rotating and boosting along the beam direction to center the jet.
2. **Crop:** Crop to a  $33 \times 33$  pixel region centered at  $(\eta, \phi) = (0, 0)$ , which captures the region with  $\eta, \phi \in (-R, R)$  for  $R = 0.4$ .
3. **Normalize:** Scale the pixel intensities such that  $\sum_{ij} I_{ij} = 1$  in the image, where  $i$  and  $j$  index over the pixels.
4. **Zero-center:** Subtract the mean  $\mu_{ij}$  of the normalized training images from each image, transforming each pixel intensity as  $I_{ij} \rightarrow I_{ij} - \mu_{ij}$ .
5. **Standardize:** Divide each pixel value by the standard deviation  $\sigma_{ij}$  of that pixel value in the normalized training dataset,  $I_{ij} \rightarrow I_{ij}/(\sigma_{ij} + r)$ . A value of  $r = 10^{-5}$  was used to suppress noise.

Steps 4 and 5 are additional pre-processing steps, not used in [12]. These new steps are often used in machine learning applications. We apply them here to put all the input pixels on an equal footing and allow the algorithm to more efficiently learn features to discriminate between classes. An improvement in performance was found after performing these additional pre-processing steps. Figure 1 shows the average centered jet images for quark jets and gluon jets before and after these two new pre-processing steps.

In addition, we implemented another useful pre-processing step, called *data augmentation* [46]: for each jet image, its three reflections about the horizontal and/or vertical axes as well as its four translations vertically or horizontally by 1 pixel were added to the dataset. These transformations enforce the discrete symmetries of the configuration and make the model robust to possible centering issues. Though our samples were not statistics limited (generating more events is relatively cheap), such an approach may be helpful in circumstances where one cannot generate more samples (e.g. with real data or with full simulation). It also helps the network to learn invariance under certain symmetries. An additional possible data augmentation would be to include additional soft particles. These could represent uncertainties in underlying event modeling or pileup and would make the model more robust to noise. We did not explore this additional augmentation step in this work.

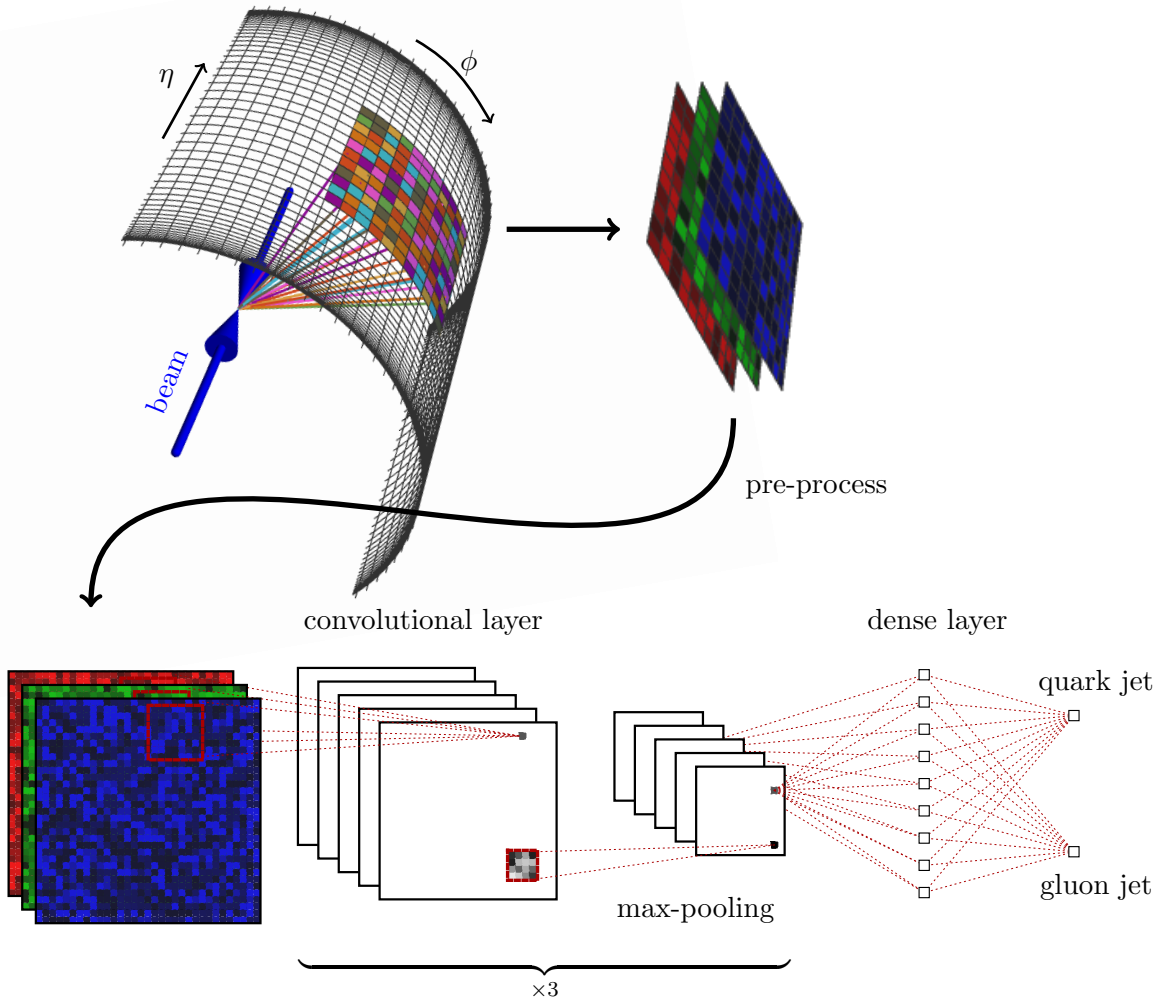


**Figure 1:** The average jet images for 200 GeV Pythia gluon jets (top) and quark jets (bottom) shown after normalization (left) and after the zero-centering and standardization (right). Different linear color scales are used to highlight the important features of each step. On the left the quark jets have more intensity in the five core pixels whereas the gluon jets are wider. On the right, the standardization procedure illustrates that quark jets are narrower and emphasizes the softer outer radiation.

### 3.2 Network architecture

The deep convolutional network architecture used in this study consisted of three iterations of a convolutional layer with a ReLU activation and a maxpooling layer, all followed by a dense layer with a ReLU activation. To predict a binary classification between quarks and gluons, an output layer of two units with sigmoid activation is fully connected to the final dense hidden layer. An illustration of the architecture used is shown in Figure 2. The dropout rate was taken to be 0.25 after the first convolutional layer and 0.5 for the remaining layers, with spatial dropout (drop entire feature maps) used in the convolutional layers. Each convolutional layer consisted of 64 filters, with filter sizes of  $8 \times 8$ ,  $4 \times 4$ , and  $4 \times 4$ , respectively.





**Figure 2:** An illustration of the deep convolutional neural network architecture. The first layer is the input jet image, followed by three convolutional layers, a dense layer and an output layer.

The maxpooling layers performed a  $2 \times 2$  down-sampling with a stride length of 2. The dense layer consisted of 128 units.

All neural network architecture training was performed with the Python deep learning libraries Keras [47] and Theano [48] on NVidia Tesla K40 and K80 GPUs using the NVidia CUDA platform. The data consisted of the 100k jet images per  $p_T$ -bin, partitioned into 90k training images and 10k test images. An additional 10% of the training images are randomly withheld as validation data during training of the model for the purposes of hyperparameter optimization. He-uniform initialization [49] was used to initialize the model weights. The network was trained using the Adam algorithm [50] using categorical cross-entropy as a loss

function with a batch size of 128 over 50 epochs and an early-stopping patience of 2 to 5 epochs.

Only moderate optimization of the network architecture and minimal hyperparameter-tuning were performed in this study. This optimization included exploration of different optimizers (Adam, Adadelta, RMSprop), filter sizes, number of filters, activation functions (ReLU, tanh), and regularization (dropout,  $L_2$ -regularization), though this exploration was not exhaustive. Further systematic exploration of the space of architectures and hyperparameter values, such as with Bayesian optimization using Spearmint [51], might increase the performance of the deep neural network.

### 3.3 Jet images in color

All implementations of the jet images machine learning approach that we know of take as the input image a grid where the input layer contains the pre-processed energy or transverse momentum in a particular angular region. This can be thought of as a grayscale image, with only intensity in each pixel and all color information discarded. In computer vision one can do better by training on color images, with red, green and blue intensities treated as separate input layers, also known as channels. Thus, it is natural to try to apply some methods for processing color to physics applications.

For particle physics, there are many ways the calorimeter deposits can be partitioned. One could try to identify the actual particles: have one channel for protons, one for neutrons, one for electrons, one for  $\pi^+$  particles, one for  $K_L$ 's, etc. Although it is not yet possible to completely separate every type of metastable particle, advances in experimental techniques, such as particle flow [52], indicate that this may not be too unrealistic. However, it is also not clear that having 15 color channels would help and training with so many input channels would be much slower. There are many options for a smaller set of channels. For example, one could consider one channel for hadrons and one for leptons, or channels for positively charged, neutral and negatively charged particles. To be concrete, in this study we take three input channels:

**red** = transverse momenta of charged particles

**green** = the transverse momenta of neutral particles

**blue** = charged particle multiplicity

Each of these observables is evaluated on each image pixel. All channels of the image undergo the following pre-processing: the images are normalized such that the sum of the red and green channels is one; the zero centering and standardization are done for each pixel in each channel according to  $I_{ij}^{(k)} \rightarrow (I_{ij}^{(k)} - \mu_{ij}^{(k)})/(\sigma_{ij}^{(k)} + r)$ . Here,  $I_{ij}^{(k)}$  is the intensity of pixel  $ij$  in channel  $k$  of an image, and  $\mu_{ij}^{(k)}$  and  $\sigma_{ij}^{(k)}$  are the respective mean and standard deviation of pixel  $ij$  in channel  $k$  in the training data.

The network architecture is designed to respect the overlay of the different color images. That is, every image channel feeds into the same units in the network and the weights from

each channel are allowed to vary independently. In other words, in the first convolutional layer instead of using an  $8 \times 8$  filter with 64 weights, we use an  $8 \times 8 \times 3$  filter with 192 weights. The number of units in the first layer is the same as with grayscale inputs, and the rest of the network architecture is the same with or without colored jet image inputs.

## 4 Fisher jets and a look inside the networks

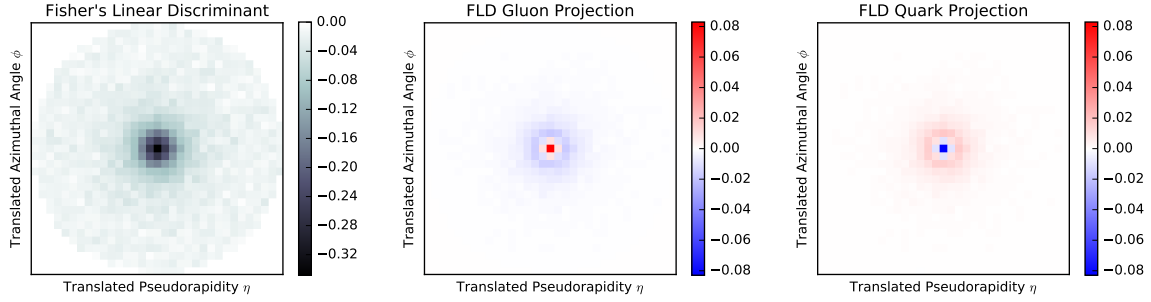
One potentially useful output of a convolutional neural network (other than the network itself) is the learned filters. These filters can display features of the image trained at different levels of resolution. In facial recognition applications, for example, one filter might pick up on noses, another on eyes, and so on.

If the entire network just had one unit, connected to all the inputs by different weights, these weights themselves would form an image and the network would act by taking the inner product of the input image and the weight image. Such a procedure, with an appropriate loss function, is equivalent to Fisher’s Linear Discriminant (FLD), which was applied to jet images in [10]. This weight image is the simplest example of a filter, and so we first find the FLD for our quark/gluon samples using the Python machine learning package scikit-learn [53]. To be more precise, the FLD method determines a Fisher jet image  $F$  that maximizes a separation by the discriminant  $D$  defined on a (grayscale) jet image  $I$  by:

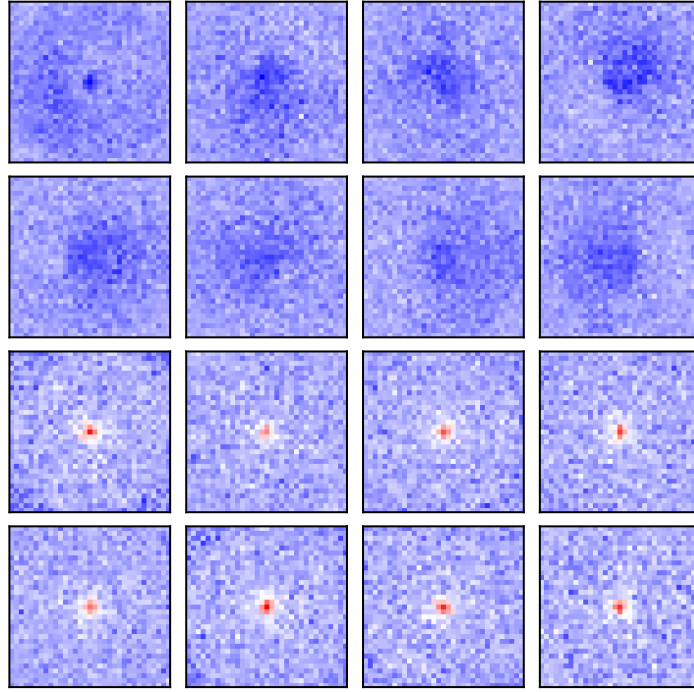
$$D[I] = \sum_{ij} F_{ij} I_{ij}. \quad (4.1)$$

Without applying the zero-centering and standardization in the pre-processing, regularization must be applied to the FLD as in [10] to prevent overfitting and reduce the sensitivity of the Fisher jet to the noisy outer bits of the jet images. However, after the full pre-processing of the jet images, no regularization is necessary to arrive at a sensible Fisher jet. An additional pre-processing step of a log transformation  $I_{ij} \rightarrow \log(1 + I_{ij}/r')$  with  $r' = 10^{-3}$  before the zero-centering was found to significantly improve the FLD performance, though this log transformation was not found to be necessary for training the deep neural networks. The Fisher jet resulting from the FLD analysis with the additional log transformation on 200 GeV jet images and its pixel-wise product with the average quark and gluon jet images are shown in Figure 3. The discriminating performance of the FLD is shown and compared to various jet variables and the deep convolutional network in Table 1 and Figure 5.

The next step up from the FLD is a shallow dense neural network, consisting of one dense layer and two output sigmoid units. This is a natural generalization of the FLD analysis. The additional units allow the network to learn more discriminating features. We trained a single-layer network consisting of a 16 fully-connected units with a ReLU activation, an L2-regularization parameter of  $10^{-7}$  on all the weights and bias terms, and a dropout rate of 0.25. The log transformation is applied to the inputs, as it was found to increase the performance of shallow networks. The 16 sets of  $33 \times 33$  weights corresponding to the observables that this model learns are shown in Figure 4. The observables generally fall into two classes:



**Figure 3:** (left) The weight vector of Fisher’s linear discriminant shown as a Fisher jet image, trained on 200 GeV Pythia jets with the additional log transformation step included in the pre-processing. The average standardized 200 GeV quark jet image (middle) and gluon jet image (right) shown after being projected with the Fisher jet.



**Figure 4:** The 16 sets of  $33 \times 33$  weights learned by the shallow dense neural network with 16 units on pre-processed 200 GeV Pythia jets. Red and blue indicate positive and negative intensity, respectively. The images are ordered from smallest to largest by the sum of their weights. The learned observables fall into two categories: observables sensitive to the large-scale jet shape and observables sensitive to the  $p_T$  near the center of the image. The first eight observables pick up on geometric moments of the jet shape, including an annular structure in the first image and several left/right asymmetric structures.

those sensitive to large geometric moments of the jet shape and those that are sensitive to the transverse momenta close to the core of the jet. The discriminating performance of this shallow dense network is compared to jet variables and the convolutional network in Table 1.

## 5 Network performance

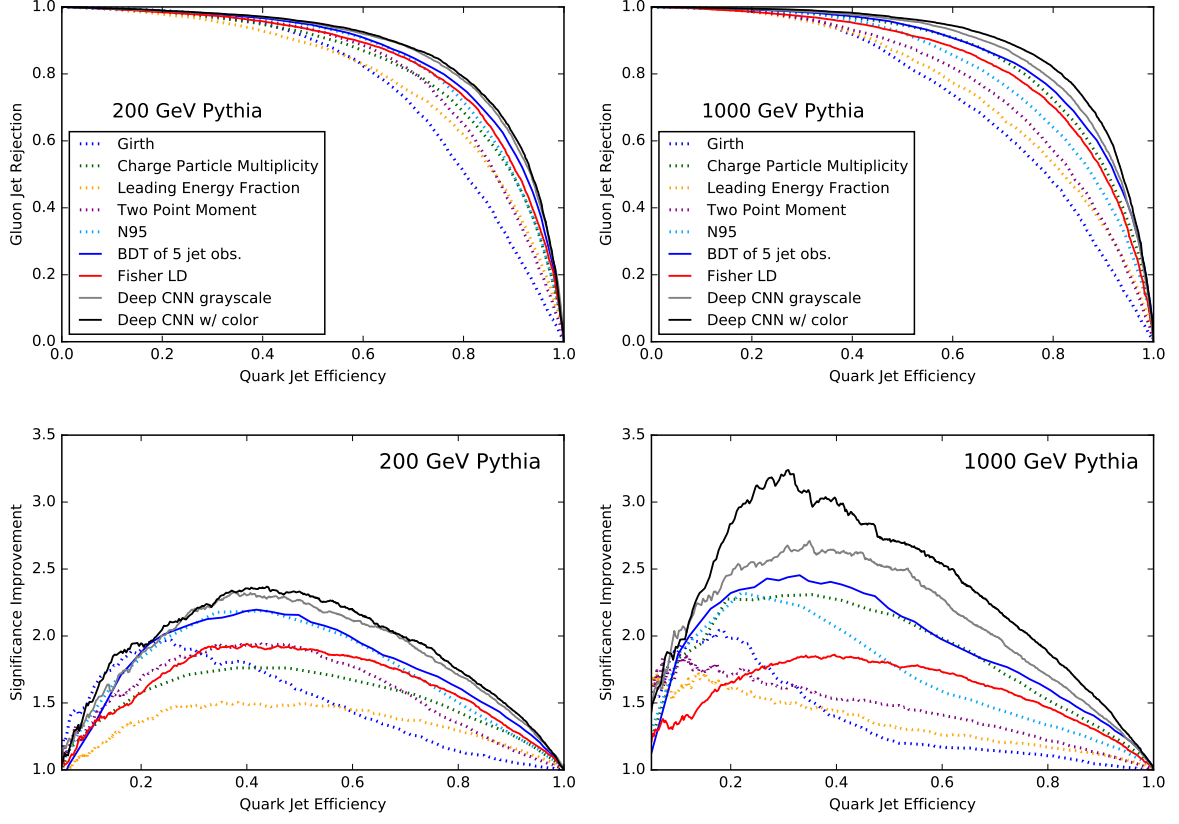
A thorough study of physics-motivated jet variables for quark/gluon discrimination was performed in [24], where continuous shape variables such as the jet girth and two-point moment and discrete variables such as the charged particle multiplicity were considered. To compare the performance of the neural networks trained on jet images to that of physics-motivated variables, the following five jet observables were considered:

- Girth:  $\sum_i p_T^i r_i / p_T^{\text{jet}}$ , where  $r_i = \sqrt{\Delta\phi_i^2 + \Delta\eta_i^2}$ . The sum is taken over the pixels in the image to account for the discretization of the detector.
- Charged Particle Multiplicity (CPM): The number of charged final-state particles in the jet. We did not apply any pixelation or detector simulation to this observable.
- Two-Point Moment [54]:  $\sum_i \sum_j p_T^i p_T^j r_i^\beta / (p_T^{\text{jet}})^2$ , where the value  $\beta = 0.2$  is used. The sum is taken over the pixels in the image to account for the discretization of the detector.
- $x_{\text{max}}$  [55]: The highest fraction of the total  $p_T$  contained in a single pixel.
- $N_{95}$  [17]: The minimum number of pixels which contain 95% of the total  $p_T$  of the jet.

The jet variable  $N_{95}$  was introduced (as  $N_{90}$ ) for quark/gluon discrimination by [17] in 1991, where a framework very similar to jet images was also introduced. In [17],  $N_{95}$  was found to be a single variable which outperformed neural networks at quark/gluon discrimination at that time, and we find that it is the physics-motivated observable with the best performance in several cases. Optimizing over the fraction of the jet  $p_T$  to consider,  $N_{95}$  performs better than  $N_{90}$  for the samples considered in this study. Deep convolutional nets notably outperform this variable, indicating an advantage from the deep learning with jet images approach over previous uses of neural networks for quark/gluon discrimination.

We measure the discrimination power of an observable by the lowest achievable gluon acceptance efficiency  $\varepsilon_g$  at a given quark acceptance efficiency  $\varepsilon_q$ . The performance can be visualized through a receiver operating characteristic (ROC) curve, which plots  $1 - \varepsilon_g$  as a function of  $\varepsilon_q$ . An alternative visualization is the significance improvement characteristic (SIC)  $\varepsilon_q / \sqrt{\varepsilon_g}$ . A SIC curve has the advantage of being closely connected to the improvement in signal over background discrimination power in a collider physics application, and also exhibits a nontrivial maximum (at some  $\varepsilon_q$ ) which gives an unbiased measure of the relative performance of different discriminants [6].

The ROC and SIC curves of the jet variables and the deep convolutional network on 200 GeV and 1000 GeV Pythia jets are shown in Figure 5. The quark jet classification



**Figure 5:** (top) ROC and (bottom) SIC curves of the FLD and the deep convolutional network trained on (left) 200 GeV and (right) 1000 GeV Pythia jet images with and without color compared to baseline jet observables and a BDT of the five jet observables.

efficiency at 50% quark jet classification efficiency for each of the jet variables and the CNN are listed in Table 1. To combine the jet variables into more sophisticated discriminants, a boosted decision tree (BDT) is implemented with scikit-learn. The convolutional network outperforms the traditional variables and matches or exceeds the performance of the BDT of all of the jet variables. The performance of the networks trained on images with and without color is shown in Figure 6.

### 5.1 Colored Jet Images

The benchmarks in the previous section were compared to the jet images with and without color, where the three color channels correspond to separating out the charge and multiplicity information as described in Section 3.3. Figure 6 shows the SIC curves of the neural network performances with and without color on Pythia jet images. For the 100 GeV and 200 GeV images, only small changes in the network performance were observed by adding in color of this form. For the 500 GeV and 1000 GeV jet images, performance increases were consistently

<b>Gluon Jet Efficiency (%) at 50% Quark Jet Acceptance</b>	1000 GeV Pythia	1000 GeV Herwig	200 GeV Pythia	200 GeV Herwig
Deep CNN with Color	3.4	18.6	4.6	16.5
Deep CNN without Color	4.0	22.2	4.8	16.4
Shallow Dense Network	6.0	23.3	5.5	18.4
Fisher’s Linear Discriminant	7.4	24.1	6.3	17.7
BDT of all jet variables	5.2	21.0	5.2	16.4
Girth $\times$ CPM	9.6	31.8	6.7	20.5
$N_{95}$	7.4	26.9	6.1	19.6
Charged Particle Multiplicity (CPM)	5.7	20.4	7.4	20.4
Two Point Moment	11.3	26.8	6.9	17.4
Girth	16.9	31.8	9.8	23.4
Leading energy fraction $x_{\max}$	14.1	28.1	11.1	21.1

**Table 1:** Gluon efficiencies at 50% quark acceptance for 200 GeV and 1000 GeV Pythia and Herwig jets using  $33 \times 33$  images. Girth  $\times$  CPM is the product of these two observables, as motivated in [24].

observed by adding color to the jet images. From Table 1, one sees that the improvement at high  $p_T$  from using multiple channels occurs with both Pythia and Herwig samples.

Alternative definitions of color, including positive, negative, and neutral  $p_T$ -channels, were considered and the present color definition was found to facilitate the best network performance within the considered architectures.

## 5.2 Merge layers

One natural question about the neural network is whether it has learned information equivalent to basic jet variables such as CPM. To approach this question, jet observables were included as additional inputs to the model through a Keras merge layer. The jet observables were fed into a two-layer dense network with 128 units per layer. The output of the second dense layer was then merged with the output of the third convolutional layer in the deep CNN, with the remainder of the CNN unchanged. If one observes a significant improvement in performance when the network is given access to the jet variable, it would indicate that the network did not learn information equivalent to that jet variable.

Figure 7 shows the SIC curves of the neural network performance on jets without color trained with additional inputs of zero,  $N_{95}$ , or CPM. Modest improvement in performance is found by adding CPM as an additional input at 500 GeV and 1000 GeV, whereas little or no improvement is found by adding  $N_{95}$  as an additional input. This behavior is indication that the network is learning geometric observables which contain information similar to  $N_{95}$ .



Image Size	$13 \times 13$	$23 \times 23$	$33 \times 33$	$43 \times 43$
<b>Gluon Jet Efficiency (%) at 50% Quark Jet Acceptance</b>	6.1	5.7	5.5	5.7

**Table 2:** Gluon jet efficiencies at 50% quark jet acceptance for 200 GeV Pythia jets using a simplified CNN (see text) applied to various image sizes. The deep CNN was applied to the  $43 \times 43$  color images and the gluon jet efficiency at 50% quark jet acceptance was 5.0% (slightly worse than the 4.6% for  $33 \times 33$  images in Table 1).

### 5.3 Image-size dependence

Additional pixel sizes for the jet images were also considered. As the deep CNN architecture illustrated in Figure 2 cannot be applied to significantly smaller input sizes due to the maxpooling layers, a simpler architecture was used: a single convolutional layer with 24 filters of size  $6 \times 6$ , a  $2 \times 2$  maxpooling, and an L2-regularization parameter of  $10^{-6}$ . The log transformation was included in the pre-processing for training this shallow network. The simplified networks were trained in the same way as the deep convolutional neural network. Table 2 shows the gluon jet efficiency at 50% quark jet acceptance for this network trained on 200 GeV Pythia jets discretized into  $13 \times 13$ ,  $23 \times 23$ ,  $33 \times 33$ , and  $43 \times 43$  grid sizes. The performance is robust over different pixelization schemes, with a decrease in performance for the smallest  $13 \times 13$  pixelization.

We also tried the full deep convolutional neural network with  $43 \times 43$  color input images, finding  $\varepsilon_g = 5.0\%$  at  $\varepsilon_q = 50\%$ . This is slightly worse than the  $\varepsilon_g = 4.6\%$  found for the same sample produced using  $33 \times 33$  pixel inputs.

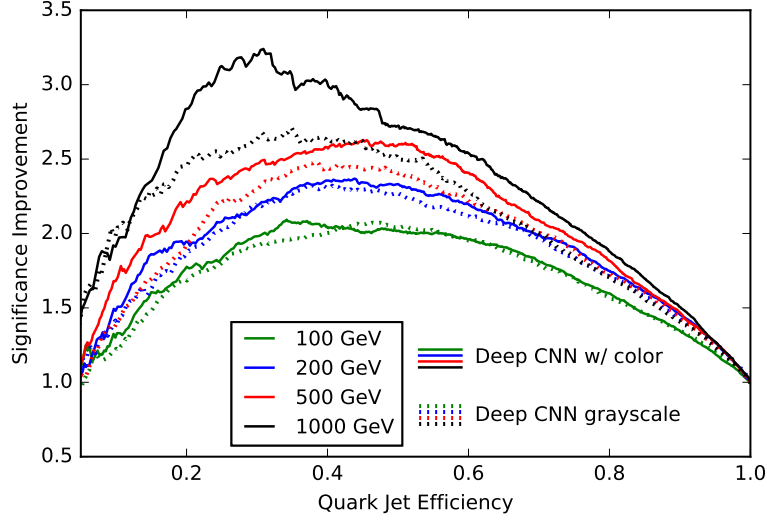
### 5.4 Event generator dependence

Many studies have shown that the two event generators Pythia and Herwig produce significantly different quark and gluon jets [24, 26, 27]. For example, the ATLAS collaboration performed a study on light-quark and gluon jet discrimination, comparing the performance of discriminants on Pythia- and Herwig-generated events to their performance on data. Quark jets and gluon jets were found to be easier to distinguish in Pythia and harder to distinguish in Herwig. The performance of the considered discriminants on data tended to be between their performance on Pythia and Herwig [27].

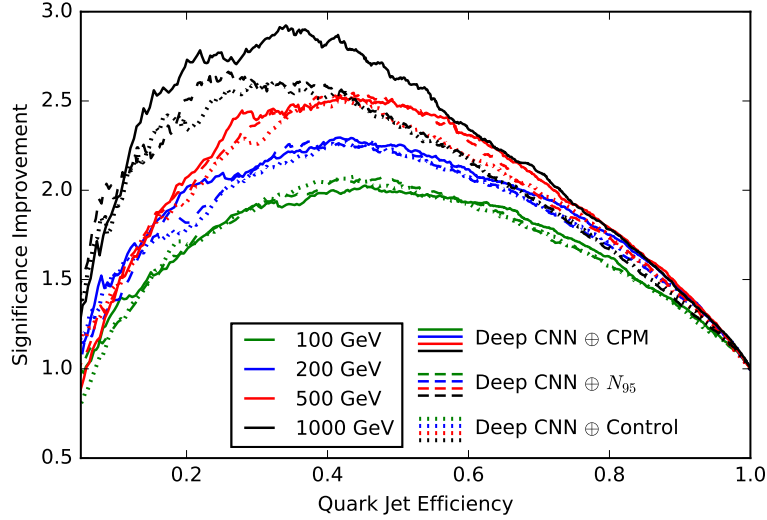
To explore the generator-dependence of our results, 200 GeV and 1000 GeV samples were generated with Pythia and Herwig. The two event generators were found to give similar quark distributions and disagree primarily on the gluon distributions. The baseline jet variables and the convolutional network all indeed have worse performance on Herwig jets than on Pythia jets. A comparison of the discrimination power of the observables between the two generators is included in Table 1.

It is interesting to consider the four possibilities of applying the convolutional networks trained on Pythia jets or Herwig jets to test samples of Pythia jets or Herwig jets. Figure 8

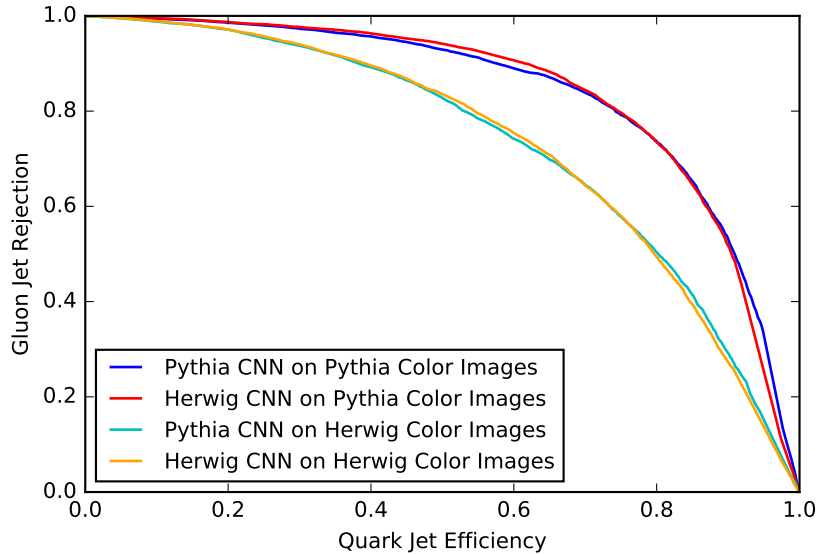




**Figure 6:** SIC curve of deep convolutional network performance on Pythia jets with color (solid) and without color (dotted). The introduction of color becomes more helpful at higher energies, with the largest improvement on the 1000 GeV jets.



**Figure 7:** SIC curve of deep convolutional network performance on Pythia jets without color with additional inputs of CPM (solid),  $N_{95}$  (dashed), and zero as a control (dotted). The spread in the SIC curves for models trained on 100 GeV and 200 GeV jets is within the typical variation, with no clear improvement from the additional variables. For models trained on 500 GeV and 1000 GeV jets, a modest improvement was seen from the introduction of CPM, though not as large as the improvement from the introduction of color.



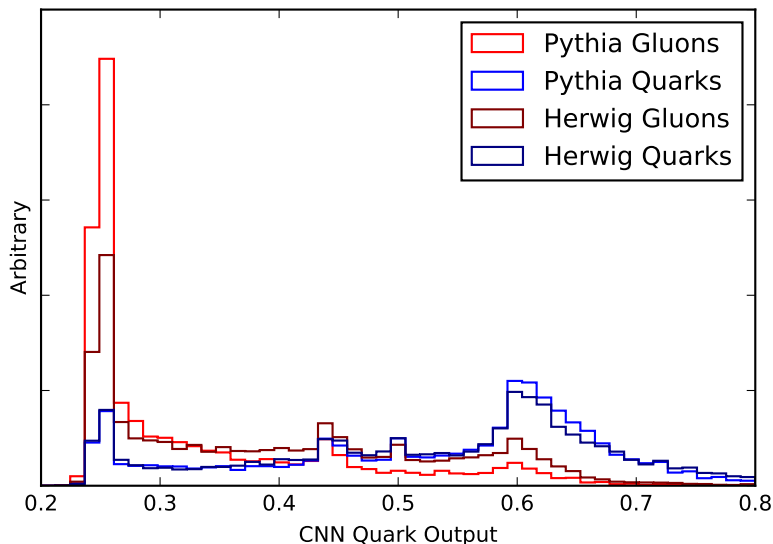
**Figure 8:** ROC curves for the Pythia- and Herwig-trained CNNs applied to 200 GeV samples generated with both of the generators. Remarkably, the network performance seems robust to which samples are used for training.

and Figure 9 show the resulting ROC curves and distributions of convolutional network outputs on the colored jet images. We find that the network is surprisingly insensitive to the generator: the convolutional network trained on Pythia jets and tested on Herwig jets has comparable performance to the convolutional network trained directly on Herwig jets and tested on Herwig jets. This insensitivity is a positive sign for being able to train the network on MC-generated jets and apply it to data robustly.

## 6 Conclusions

The ability to distinguish quark-initiated jets from gluon-initiated jets would be of tremendous practical application at colliders like the LHC. For example, many signals of beyond the standard model physics contain mostly quark jets, while their backgrounds are gluon-jet dominated. Quark/gluon jet discrimination is also extremely challenging: correlations in their radiation patterns and non-perturbative effects like hadronization are hard to disentangle. Thus this task is ideally suited for artificial intelligence.

In this paper, we have applied machine learning techniques developed for computer vision, namely deep convolutional artificial neural networks, to the quark/gluon differentiation problem. Overall, we find excellent performance of the deep networks. In particular, these networks, which use essentially no input about the physics underlying the differences between these two jet types, performs as well or better than a collection of the best physically motivated observables from other studies (see Table 1).



**Figure 9:** Distribution of Pythia-trained colored-CNN outputs for 200 GeV jets. Quark samples are blue and gluon samples are red. Pythia samples are shown in the lighter shade and Herwig samples are shown in the darker shade.

The input layer of our neural network is taken to be the transverse momentum deposited in a particular region of the detector. We preserve the locality of this energy deposition by constructing a 2 dimensional jet image, with the 2 dimensions corresponding to the surface of a cylinder, in pseudorapidity and azimuthal angle (see Figure 2). It is not completely obvious that locality in position should be helpful in quark/gluon discrimination. On the one hand, locality is clearly relevant, as infrared safety demands that observables include integrations over compact phase space regions. On the other hand, quarks and gluons are not infrared-safe objects; indeed hadronization, which depends on the color charge of the fragmenting partons, is non-local. The relevance of locality is essentially a consequence of the collinear singularity in the QCD splitting functions and of soft color coherence. Convolutional network architectures are structured to take advantage of the local information in the input while being able to learn non-local observables.

In addition to using the overall energy deposited in a local region, our network also exploits correlations in the particle charge and particle multiplicity. To use this information, we treat the input as a colored jet image. The red image is the transverse momenta in charged particles, the blue image is the transverse momenta of neutral particles, and the green image is the (local) charged particle multiplicity. Rather than colors, one can also think of these extra inputs as making our image 3 dimensional, with the third dimension only 3 pixels deep. The additional information gives a marginal improvement at 200 GeV, but for very high  $p_T$  jets, the improvement is substantial. This is probably due to the overall higher multiplicity at higher energies and correspondingly enhanced discrimination power

of observables that exploit this information (this can also be seen in the improved relative performance of charged-particle multiplicity at high  $p_T$  in Table 1).

A standard criticism of neural networks is that they are only as smart as the data used to train them is accurate: if the simulations are poor, it’s garbage-in-garbage-out. For quark and gluon jet discrimination, this criticism must be taken seriously, as the simulations are known to be poor. Indeed, the two standard generators, Pythia and Herwig, differ substantially in their jet simulations with Herwig quark and gluon jets substantially more similar to each other than Pythia quark and gluon jets. We confirm this observation, as our NNs are substantially worse at correctly labelling Herwig jets than Pythia jets. However, somewhat surprisingly, we find that the network performance is the same whether trained on a Pythia or Herwig sample. In other words, the networks may be picking up on underlying physical distinctions between the jets which are similar in the two simulations, just to a lesser degree. Essentially, the NN, when trained on either sample, is acting much like a physically-motivated observable. This suggests that the NN may be more trustworthy when acting on data than the simulations used to train them.

A second criticism of neural networks used in physics is that they are black boxes. They could be picking up on unphysical features of the simulations. While it is impossible to completely refute this criticism, it is certainly possible to explore what the networks are doing. For example, the convolutional network layers comprise filters which can be examined. An example of such filters is given in Fig. 4 which shows how the NN is picking up on different elements of the radiation distribution. Also we can add merge layers, as described in Section 5.2, which allow us to determine if a particular observable has been learned by the network. In fact, any multivariate technique, such as a boosted decision tree constructed from ten well-motivated observables, can be attacked with the black-box criticism. The truth is that because of the subtlety of the identification tasks we need these multivariate methods to perform, some lack of low-level understanding of what these methods are doing is unavoidable. Not understanding them does not mean the methods do not work.

As particle colliders push to higher and higher luminosity, and as the signals we search for become ever more complex and subtle, the reliance on machine learning will eventually be inevitable. Indeed, machine learning has a bright future in particle physics, as developments in processing power, computer science and in event-generator simulations, give us more and more reason to use these methods. Moreover, artificial intelligence has the advantage of not being limited by human creativity. For example, a comprehensive quark/gluon discrimination study in [24] classified useful observables as being either “shapes” or “counts”. However, another study [17] found that  $N_{90}$ , a hybrid shape/count observable, works exceedingly well. One conclusion of our study is that deep neural networks can perform as well as any of the considered physical observables, or any combination thereof. Although we are not recommending that these networks be used on data yet (since the simulations on which they are trained cannot yet be trusted), our study does suggest that artificial intelligence will eventually play an essential role in quark/gluon jet discrimination and probably many other tasks in particle physics.

## Acknowledgments

The authors would like to thank Jesse Thaler, Christopher Frye, and Frédéric Dreyer for useful discussions. PTK and EMM would like to thank the MIT Physics Department for its support. MDS was supported in part by the U.S. Department of Energy under grant DE-SC0013607. The computations in this paper were run on the Odyssey cluster supported by the FAS Division of Science, Research Computing Group at Harvard University.

## References

- [1] ATLAS Collaboration, *A neural network clustering algorithm for the ATLAS silicon pixel detector*, 2014 *JINST* **9** P09009 [[arXiv:1406.7690](#)].
- [2] ATLAS Collaboration, *Performance of  $b$ -Jet Identification in the ATLAS Experiment*, 2016 *JINST* **11** P04008 [[arXiv:1512.01094](#)].
- [3] M. Tosi, *Performance of Tracking,  $b$ -tagging and Jet/MET reconstruction at the CMS High Level Trigger*, *J. Phys. Conf. Ser.* **664** (2015).
- [4] CMS collaboration, *Performance of  $\tau$ -lepton reconstruction and identification in CMS*, 2012 *JINST* **7** P01001 [[arXiv:1109.6034](#)].
- [5] NNPDF Collaboration, R.D. Ball et al., *Parton distributions for the LHC Run II*, *JHEP* **04** (2015) 040 [[arXiv:1410.8849](#)].
- [6] J. Gallicchio, J. Huth, M. Kagan, M. D. Schwartz, K. Black and B. Tweedie, *Multivariate discrimination and the Higgs +  $W/Z$  search*, *JHEP* **04** (2011) 069 [[arXiv:1010.3698](#)].
- [7] T. Maggipinto et al., *Role of neural networks in the search of the Higgs boson at LHC*, *Phys. Lett. B* **409** (1997) 517 [[arXiv:9705020](#)].
- [8] CMS collaboration, *Search for supersymmetry in events with opposite-sign dileptons and missing transverse energy using an artificial neural network*, *Phys. Rev. D* **87** (2013) 072001 [[arXiv:1301.0916](#)].
- [9] P. Baldi, P. Sadowski and D. Whiteson, *Enhanced Higgs Boson to  $\tau^+\tau^-$  Search with Deep Learning*, *Phys. Rev. Lett.* **114** (2015) 111801 [[arXiv:1410.3469](#)].
- [10] J. Cogan, M. Kagan, E. Strauss and A. Schwartzman, *Jet-Images: Computer Vision Inspired Techniques for Jet Tagging*, *JHEP* **02** (2015) 118 [[arXiv:1407.5675](#)].
- [11] L. G. Almeida, M. Backović, M. Cliche, S. J. Lee and M. Perelstein, *Playing Tag with ANN: Boosted Top Identification with Pattern Recognition*, *JHEP* **1507** (2015) 086 [[arXiv:1501.05968](#)].
- [12] L. de Oliveira, M. Kagan, L. Mackey, B. Nachman and A. Schwartzman, *Jet-images – deep learning edition*, *JHEP* **07** (2016) 069 [[arXiv:1511.05190](#)].
- [13] P. Baldi, K. Bauer, C. Eng, P. Sadowski and D. Whiteson, *Jet Substructure Classification in High-Energy Physics with Deep Neural Networks*, *Phys. Rev. D* **93** (2016) 094034 [[arXiv:1603.09349](#)].

- [14] D. Guest, J. Collado, P. Baldi, S. C. Hsu, G. Urban and D. Whiteson, *Jet Flavor Classification in High-Energy Physics with Deep Neural Networks*, *Phys. Rev. D* **94** (2016) 112002 [[arXiv:1607.08633](#)].
- [15] J. Barnard, E.N. I Dawe, M.J. Dolan, and N. Rajcic, *Parton Shower Uncertainties in Jet Substructure Analyses with Deep Neural Networks* [[arXiv:1609.00607](#)].
- [16] A. Krizhevsky, I. Sutskever, and G.E. Hinton, *Imagenet classification with deep convolutional neural networks*, in the proceedings of *Neural Information Processing Systems* (NIPS 2012), December 3-8, Lake Tahoe, U.S.A (2012).
- [17] J. Pumplin, *How to tell quark jets from gluon jets*, *Phys. Rev. D* **44** (1991) 2025.
- [18] L. Lönnblad, C. Peterson, and T. Rönkvallsson, *Finding gluon jets with a neural trigger*, *Phys. Rev. Lett.* **65** (1990) 1321.
- [19] OPAL Collaboration, *A Study of differences between quark and gluon jets using vertex tagging of quark jets*, *Z. Phys. C* **58** (1993) 387.
- [20] OPAL Collaboration, *A direct observation of quark-gluon jet differences at LEP*, *Phys. Lett. B* **265** (1991) 462.
- [21] D. Ferreira de Lima, P. Petrov, D. Soper, and M. Spannowsky, *Quark-Gluon tagging with Shower Deconstruction: Unearthing dark matter and Higgs couplings* [[arXiv:1607.06031](#)].
- [22] A.J. Larkoski, J. Thaler, and W.J. Waalewijn, *Gaining (mutual) information about quark/gluon discrimination*, *JHEP* **11** (2014) 129 [[arXiv:1408.3122](#)].
- [23] B. Bhattacharjee, S. Mukhopadhyay, M.M. Nojiri, Y. Sakaki, and B.R. Webber, *Associated jet and subjet rates in light-quark and gluon jet discrimination*, *JHEP* **04** (2013) 090 [[arXiv:1501.04794](#)].
- [24] J. Gallicchio and M. D. Schwartz, *Quark and gluon jet substructure*, *JHEP* **04** (2013) 090 [[arXiv:1211.7038](#)].
- [25] J. Gallicchio and M. D. Schwartz, *Quark and gluon tagging at the LHC*, *Phys. Rev. Lett.* **107** (2011) 172001 [[arXiv:1106.3076](#)].
- [26] J.R. Andersen et al., *Les Houches 2015: Physics at TeV Colliders Standard Model Working Group Report* [[arXiv:1605.04692](#)].
- [27] ATLAS Collaboration, *Light-quark and gluon jet discrimination in pp collisions at  $\sqrt{s} = 7$  TeV with the ATLAS detector*, *Eur. Phys. J. C* **74** (2014) 3023 [[arXiv:1405.6583](#)].
- [28] C. Frye, A.J. Larkoski, M.D. Schwartz and K. Yan, *Factorization for groomed jet substructure beyond the next-to-leading logarithm*, *JHEP* **07** (2016) 064, [[arXiv:1603.09338](#)].
- [29] R. Collobert and J. Weston, *A unified architecture for natural language processing: deep neural networks with multitask learning*, in the proceedings of the 25<sup>th</sup> *International Conference on Machine Learning*, July 5-9, Helsinki, Finland (2008).
- [30] P. Baldi, P. Sadowski, and D. Whiteson, *Searching for exotic particles in high-energy physics with deep learning*, *Nature Commun.* **5** (2014) 4308 [[arXiv:1402.4735](#)].
- [31] S. Dieleman, K.W. Willet, and J. Dambre, *Rotation-invariant convolutional neural networks for galaxy morphology prediction*, *Mon. Roy. Astron. Soc.* **450** (2015) 1441 [[arXiv:1503.07077](#)].

- [32] M.A. Nielsen, *Neural Networks and Deep Learning*, Determination Press, U.S.A (2015).
- [33] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, U.S.A. (2016).
- [34] X. Glorot, A. Bordes, and Y. Bengio, *Deep sparse rectifier neural networks*, proceedings of the 14<sup>th</sup> *International Conference on Artificial Intelligence and Statistics*, April 11-13, Ft. Lauderdale, U.S.A. (2011).
- [35] K. Hornik, M. Stinchcombe, and H. White, *Multilayer feedforward networks are universal approximators*, *Neural Netw.* **2** (1989) 359.
- [36] J. Schmidhuber, *Deep learning in neural networks: an overview*, *Neural Netw.* **61** (2015) 85 [[arXiv:1404.7828](#)].
- [37] C. Szegedy et al., *Going deeper with convolutions*, in the proceedings *IEEE Conference on Computer Vision and Pattern Recognition*, June 7-12, Boston U.S.A. (2015) [[arXiv:1409.4842](#)].
- [38] N. Srivastava et al., *Dropout: a simple way to prevent neural networks from overfitting*, *JMLR* **15** (2014) 1929.
- [39] T. Sjostrand, et al., *An Introduction to PYTHIA 8.2*, *Comput. Phys. Commun.* **191** (2015) 159, [[arXiv:1410.3012](#)].
- [40] M. Bahr et al., *Herwig++ physics and manual*, *Eur. Phys. J. C* **58** (2008) 639 [[arXiv:0803.0883](#)].
- [41] J. Bellm et al., *Herwig 7.0 / Herwig++ 3.0 release note*, *Eur. Phys. J. C* **76** (2016) 196 [[arXiv:1512.01178](#)].
- [42] M. Cacciari, G.P. Salam, and G. Soyez, *FastJet user manual*, *Eur. Phys. J. C* **72** (2012) 1896 [[arXiv:1111.6097](#)].
- [43] M. Cacciari, G.P. Salam, and G. Soyez, *The anti- $k_t$  jet clustering algorithm*, *JHEP* **04** (2008) 063 [[arXiv:0802.1189](#)].
- [44] D. E. Kaplan, K. Rehermann, M. D. Schwartz and B. Tweedie, *Top tagging: a method for identifying boosted hadronically decaying top quarks*, *Phys. Rev. Lett.* **101** (2008) 142001 [[arXiv:0806.0848](#)].
- [45] NNPDF Collaboration, R.D. Ball et al., *A determination of parton distributions with faithful uncertainty estimation*, *Nucl. Phys.* **B 809** (2009) 1 [Erratum *ibid.* **B 816** (2009) 293] [[arXiv:0808.1231](#)].
- [46] P.Y. Simard, D. Steinkraus, and J.C. Platt, *Best practices for convolutional neural networks applied to visual document analysis*, *ICDAR* **3** (2003).
- [47] F. Chollet, *Keras*, available at [GitHub](#) (2015).
- [48] J. Bergstra et al., *Theano: a CPU and GPU math compiler in python*, in the proceedings of the 9<sup>th</sup> *Python in Science Conference*, June 28-July 3, Austin, U.S.A. (2010).
- [49] K. He, X. Zhang, S. Ren, and J. Sun, *Delving deep into rectifiers: surpassing human-level performance on imagenet classification*, in the proceedings of the *IEEE International Conference on Computer Vision*, December 11-18, Santiago, Chile (2015).
- [50] D. Kingma and J. Ba, *Adam: A method for stochastic optimization*, [arXiv:1412.6980](#).

- [51] J. Snoek, H. Larochelle, and R.P. Adams, *Practical bayesian optimization of machine learning algorithms*, in the proceedings of *Neural Information Processing Systems (NIPS 2012)*, December 3-8, Lake Tahoe, U.S.A. (2012), [arXiv:1206.2944](#).
- [52] CMS Collaboration, *Particle-flow event reconstruction in CMS and performance for jets, taus, and MET*, CMS-PAS-PFT-09-001 (2009).
- [53] F. Pedregosa et al., *Scikit-learn: machine learning in python*, *JMLR* **12** (2012) 2825.
- [54] A.J. Larkoski, G.P. Salam, and J. Thaler, *Energy correlation functions for jet substructure*, *JHEP* **06** (2013) 108 [[arXiv:1305.0007](#)].
- [55] H. Küçük, *Measurement of the inclusive-jet cross-section in proton-proton collisions and study of Quark-Gluon Jet discrimination with the ATLAS experiment at the LHC*, Dissertation, University College London, London. U.K. (2016).