Anomalous Jet Tagging via Sequence Modeling

Alan Kahn[†], Julia Gonski[†], Inês Ochoa[‡], Daniel Williams[†], Gustaaf Brooijmans[†]

[†]Nevis Laboratories, Columbia University in the City of New York, USA
[‡]Laboratory of Instrumentation and Experimental Particle Physics, Lisbon, Portugal

Abstract

This paper presents a novel method of searching for boosted hadronically decaying objects by treating them as anomalous elements of a contaminated dataset. A *Variational Recurrent Neural Network* (VRNN) is used to model jets as sequences of four-vector constituents. Coupled with a carefully considered pre-processing, the VRNN gives each jet an *Anomaly Score* that distinguishes between the substructure of signal and background jets. This is done without high level variables, making the score more robust against mass and p_T correlations when compared to traditional substructure-based methods. The Anomaly Score shows consistent performance along a wide range of signal contamination amounts, for both two- and three-pronged jet substructure hypotheses. The model is trained in an entirely unsupervised setting, opening up the possibility to train directly on data without a pre-defined signal hypothesis. Performance is evaluated on the jet level, as well as in an analysis context by searching for a heavy resonance with a final state of two boosted jets. Results demonstrate that the use of Anomaly Score as a classifier enhances signal sensitivity while retaining a smoothly falling background jet mass distribution.

1 Introduction

The frontier of high energy physics has many open questions, such as the nature of dark matter or the origin of matter-antimatter asymmetry, which are being explored by the ATLAS and CMS Experiments at the Large Hadron Collider (LHC). Since the discovery of the Higgs boson in 2012 [6, 7], no signals of new beyond the Standard Model (BSM) physics have been found. Sophisticated analysis tools utilizing novel machine learning models are key to detecting rare signals that may be missed by traditional methods.

Machine learning is being consistently integrated into new physics searches at the LHC. While many applications focus on classifying known signatures, such as top quarks or Higgs bosons, there are a number of emerging techniques aimed at the discovery of new particles. Many of these techniques use classifier models to search for particular model hypotheses, by training and evaluating a model using Monte Carlo simulation of the BSM signal. While these models show promising performance, they are subject to inaccuracies between simulated samples and data, which occur most severely in the non-perturbative regime of QCD processes.

One way to circumvent these inaccuracies is to develop a model that trains directly on data, without requiring the need for simulated inputs. Distinguishing new physics from Standard Model background at the LHC without a signal hypothesis is a novel and promising effort [15]. This paper demonstrates a way to identify BSM signals that present as anomalous substructure in jets using unsupervised, data-driven anomaly detection.

Anomaly detection refers to a process in which anomalous elements are identified within a dataset that is mostly homogenous, but contaminated with outliers. In a machine learning context, this can be done with a model that learns an underlying distribution of data points, as characterized by high-level features of the data. Such a model can then identify out-of-distribution data solely on how poorly they are represented by the learned underlying distribution. Several candidate architectures have been developed for this purpose. The examination of their features is instructive in choosing an architecture for the specific task presented here.

1.1 Autoencoders

A popular candidate architecture for anomaly detection is the *autoencoder* (AE) [2], which has been previously studied in a particle physics context [8, 11]. Autoencoders are an example of a generative model in which a network is trained to reconstruct a given input. Figure 1 shows an example of a standard AE architecture.

A key feature of autoencoders is a latent layer in the center of the architecture which is often of a lower dimensionality than the input, directly restricting the network's ability to perfectly reconstruct its input. In such a case, the network achieves its training goal best when it can represent high-level features of the input as vectors, or $codes$, in its latent space. The accuracy with which each code represents the input can be verified by decoding it, and comparing its result with the original input. In this way, the AE is considered to act as two neural networks being trained in parallel: an $encoder$ network f which acts as the map from data to the latent space, $\mathbf{z} = f(\mathbf{x})$, and a $decoder$ network g which then attempts to reconstruct the original input from its encoded representation, $\mathbf{y} = g(\mathbf{z})$. The loss function of the AE can be any function of

the form $\mathcal{L}(\mathbf{x}, \mathbf{y} = g(f(\mathbf{x})))$, which is minimal when $\mathbf{y} = \mathbf{x}$. A common choice is the *Mean Squared Error* (MSE) between the input and output of the autoencoder:

$$\mathcal{L} = |\mathbf{y} - \mathbf{x}|^2 \tag{1}$$

In the context of anomaly detection, elements which represent a small portion of a dataset will contribute less during the training process. As a result, they will be less represented by the learned codes when compared to elements belonging to the majority class of data. One can therefore expect the reconstruction of anomalous elements to be worse, placing them in the tails of the loss function's distribution after training. This principle has been explored in anomalous jet tagging, for instance by representing the jets as images [8], or as lists of constituents [11].

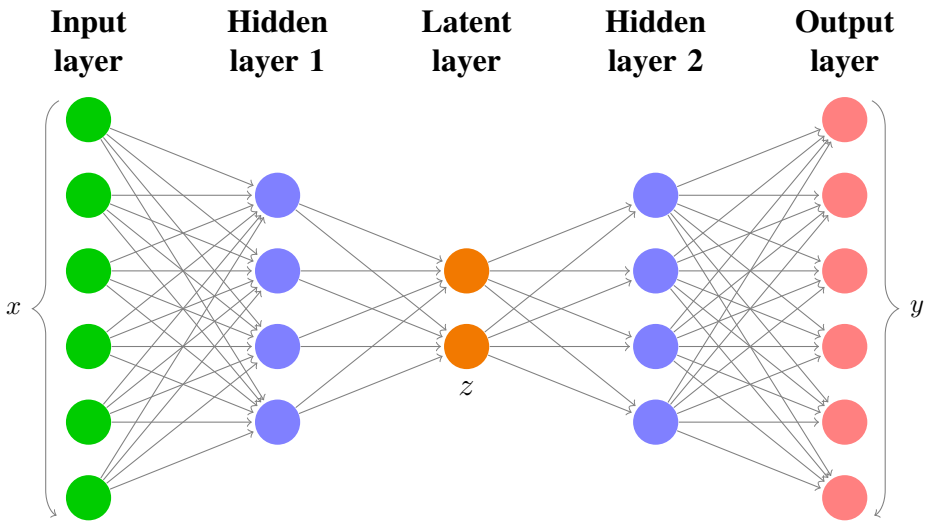

Figure 1: A standard autoencoder.

1.2 Variational Autoencoders

Variational Autoencoders (VAEs) are built on the idea of standard AEs, with the extension that they are designed to perform Bayesian inference. This assumes that observed data $\mathbf{x}$ is generated by some hidden random variable $\mathbf{z}$ whose posterior distribution $p(\mathbf{z}|\mathbf{x})$ is intractable. The goal of a VAE is to learn an approximate posterior distribution, $q(\mathbf{z}|\mathbf{x})$, through training. In this way, the latent space of the VAE then represents the approximate posterior distribution.

The architecture of a VAE is very close to that of a standard autoencoder. The main difference is a latent space that can accommodate an encoder, which maps data to a distribution in the latent space rather than a single vector. A common choice for the form of the latent space is a multivariate gaussian of diagonal covariance. In this case, the encoder can map a given input to two independent layers, each with the same dimensionality as the latent space, which together represent the means and standard deviations of the encoded gaussian distribution for each dimension respectively. Decoding then requires sampling from this resulting distribution. This can be easily performed in the case of a gaussian approximate posterior by use of the *reparametrization trick*. Instead of sampling the distribution directly, a particular value of z can be represented in the following way:

$$z = \mu + \sigma\epsilon \tag{2}$$

Here, ϵ is sampled from a unit isotropic normal distribution $\epsilon \sim \mathcal{N}(0, 1)$ [14].

The VAE performs Bayesian inference by determining the marginal likelihood, which is the result of an often intractable calculation:

$$p(x) = \int p(z)p(x|z)dz \tag{3}$$

By using Bayes' Theorem, and inserting the approximate posterior distribution $q(z|x)$, the log likelihood can be expressed in terms of two Kullback-Leibler (KL)-Divergences, one from a prior distribution $p(z)$ to the approximate posterior $q(z|x)$, and the other from the true posterior distribution $p(z|x)$ to the same approximate posterior $q(z|x)$. The remaining term is the log likelihood of data, and can be interpreted as the reconstruction accuracy of generating x from the underlying variable z.

$$\begin{aligned}
\log(p(x)) &= \mathbb{E}_Z[\log p(x)] \\
&= \mathbb{E}_Z\left[\log \frac{p(x|z)p(z)}{p(z|x)}\right] \\
&= \mathbb{E}_Z\left[\log \frac{p(x|z)p(z)}{p(z|x)}\frac{q(z|x)}{q(z|x)}\right] \\
&= \mathbb{E}_Z[\log p(x|z)] - \mathbb{E}_Z\left[\log \frac{q(z|x)}{p(z)}\right] + \mathbb{E}_Z\left[\log \frac{q(z|x)}{p(z|x)}\right] \\
&= \underbrace{\mathbb{E}_Z[\log p(x|z)]}_{\text{Reconstruction Error}} - \underbrace{\int q(z|x)\log\frac{q(z|x)}{p(z)}dz}_{D_{KL}(q(z|x)||p(z))} + \underbrace{\int q(z|x)\log\frac{q(z|x)}{p(z)}dz}_{D_{KL}(q(z|x)||p(z|x))}
\end{aligned} \tag{4}$$

While the true posterior distribution is still intractable, the KL-Divergence is by definition non-negative. Thus the first two terms of this result can be described as a lower bound on the evidence. When this lower bound is maximized, the remaining intractable KL-Divergence approaches zero, corresponding to a situation in which the reconstruction error is zero, and the approximate posterior is equivalent to the true posterior. Therefore, the negative of this lower bound is chosen as the loss function of the VAE, and is minimized through training. The reconstruction error term is chosen to be the mean-squared-error loss as used in the ordinary AE. The total loss function therefore takes the following form:

$$\mathcal{L} = -|\mathbf{y} - \mathbf{x}|^2 + D_{KL}(q(z|x)||p(z)) \tag{5}$$

For the prior, it is common to choose a unit isotropic gaussian centered at the origin, as the KL-Divergence between a gaussian approximate posterior and a gaussian prior takes on a closed form solution [9].

Variational Autoencoders provide a number of improvements over standard Autoencoders, both as a generative model [14] and as an anomaly detection tool [1]. The inclusion of a KL-Divergence term in the loss function, in addition to motivating the architecture to more appropriately model unique classes of data, exists as another discriminatory metric used to differentiate nominal and anomalous elements of the dataset. Anomalous elements are expected to have both a large reconstruction error and a large KL-Divergence when compared to nominal elements.

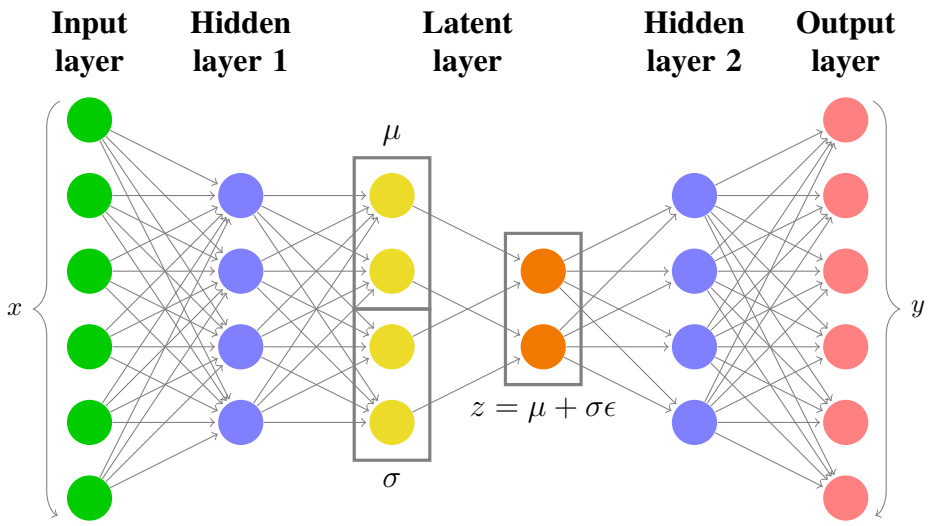

Figure 2: A Variational Autoencoder with a gaussian latent space parametrization.

While VAEs have shown promise in the task of jet-level anomaly detection, they have a number of drawbacks. Most notably, VAEs are a fixed-length architecture, and cannot accommodate a variable number of inputs. When modeling jets via their constituent four-vectors, it thus becomes necessary to only process at most N constituents, and *zero-pad* the input layer when processing a jet with a number of constituents less than N. In classifier models, this is common and benign, as the loss function depends only on the output of the network and the ground-truth that it is trying to reproduce. However, in a VAE, the input layer's neuron values are a part of its loss function (due to the MSE loss between the input and output layers). Therefore, the zero padded elements directly correlate with the value of the loss function. This introduces a direct correlation between the VAE loss and the number of constituents in the input jet which can be difficult to remove.

A recurrent architecture naturally circumvents this drawback since it is designed to accommodate inputs of varying length. In a *Recurrent Neural Network* (RNN), data is input as a sequence of features. Each feature has the same fixed dimensionality, yet the sequence itself can vary in length. The RNN is comprised of a small fixed architecture (JG there are multiple?), or *cell*, which expects as an input the fixed-length feature at each element, or *time-step*, in the sequence. While processing the sequence, the RNN updates a *hidden state* at each time-step, which is carried over and accessed by the cell during the following time-step. The hidden state stores a long-term representation of information within the sequence, and is the key feature allowing RNNs to process sequential data of varying length. The RNN cell then acts as an encoder-decoder architecture which inputs the current time-step's feature and hidden state, and outputs an updated hidden state, along with an output feature if desired. In the interest of performing anomaly detection using a recurrent architecture, the model in this study has been chosen to be one which combines the recurrent property of RNNs with the VAE's ability to perform variational inference.

2 Variational Recurrent Neural Network

The Variational Recurrent Neural Network (VRNN) used in this study is a sequence modeling architecture which replaces the encoder-decoder step of a traditional RNN with a VAE. An illustration of one VRNN cell can be seen in Figure 3. In this model, the VAE's input at each time-step is given as the vector $x(t)$, which is then encoded and decoded into an output vector $y(t)$ which can be compared to $x(t)$ via the reconstruction loss. The ϕ_x and ϕ_z layers represent *feature-extracting layers*, which are interpreted as learned representations of the features of the input $x(t)$ and the encoded latent space distribution $z(t)$, respectively. After each time-step, the hidden state is updated via a recurrence relation, in which the current hidden state $h(t-1)$ and the current set of extracted features ϕ_x and ϕ_z produce an updated hidden state $h(t)$ via the following equation [5]:

$$h(t) = f(\phi_x, \phi_z, h(t-1)) \tag{6}$$

Performing this particular step is the primary function of traditional RNN architectures such as Long Short-Term Memory Networks (LSTMs) [12] and Gated Recurrent Units (GRUs) [4].

The VAE present in each cell of the VRNN notably differs from conventional VAEs in the following ways:

1. The encoder and decoder are conditioned on the current time-step's hidden state. This is represented by the concatenation operation between the hidden state $h(t-1)$ and the feature-extraction layers ϕ_x and ϕ_z.
2. The prior from which the KL-Divergence is computed is no longer a unit gaussian at the origin, but rather a multivariate gaussian whose means and variances in each dimension are determined from the current time-step's hidden state.

JG: Introduce why you're digging into the prior/approx posterior descriptions In more detail, each time-step's latent space prior distribution parameters μ_t and σ_t are functions of the current time-step's hidden state:

$$z_t \sim \mathcal{N}(\mu_t, \sigma_t), \text{ where } \mu_t, \sigma_t = f^{prior}(h_{t-1}) \tag{7}$$

Similarly, the latent space approximate posterior is defined by parameters μ and σ which are functions of the input's extracted features ϕ_x and the hidden state h_{t-1}

$$z \sim \mathcal{N}(\mu, \sigma), \text{ where } \mu, \sigma = f^{post.}(\phi_x, h_{t-1}) \tag{8}$$

The generated output is then decoded from features extracted from the latent space distribution $\phi_z = f(z)$, while also being conditioned on the hidden state

$$y(t) = f^{dec}(\phi_z, h(t-1)) \tag{9}$$

A loss for each time-step $\mathcal{L}(t)$ can then be computed by incorporating both the reconstruction error between the input constituent $x(t)$ and generated output constituent $y(t)$, as well as the KL-Divergence between the approximate posterior z and the learned prior z_t. A constant λ is also included which weights the KL-Divergence term's contribution to the loss.

$$\mathcal{L}(t) = |\mathbf{y}(t) - \mathbf{x}(t)|^2 + \lambda D_{KL}(z||z_t) \tag{10}$$

An overall loss $\mathcal{L}$ over the sequence is then computed by averaging the individual time-step losses over the length of the sequence N

$$\mathcal{L} = \frac{\mathcal{L}(t)}{N} \tag{11}$$

This loss function takes the same role as the VAE's loss function, acting both as an appropriate means of optimizing the architecture as well as a discriminatory quantity between nominal and anomalous elements of the dataset.

The inclusion of a learned, time-dependent prior distribution is an important component of the VRNN architecture. Without this feature, the decoder network would only be able to access information about the current time-step from the hidden state, and the loss function would motivate the posterior distributions for each time-step to be identical. As a result, this allows the VRNN the flexibility to model complex structured sequences with high variability, as is expected from a jet represented by a sequence of constituent four-vectors.

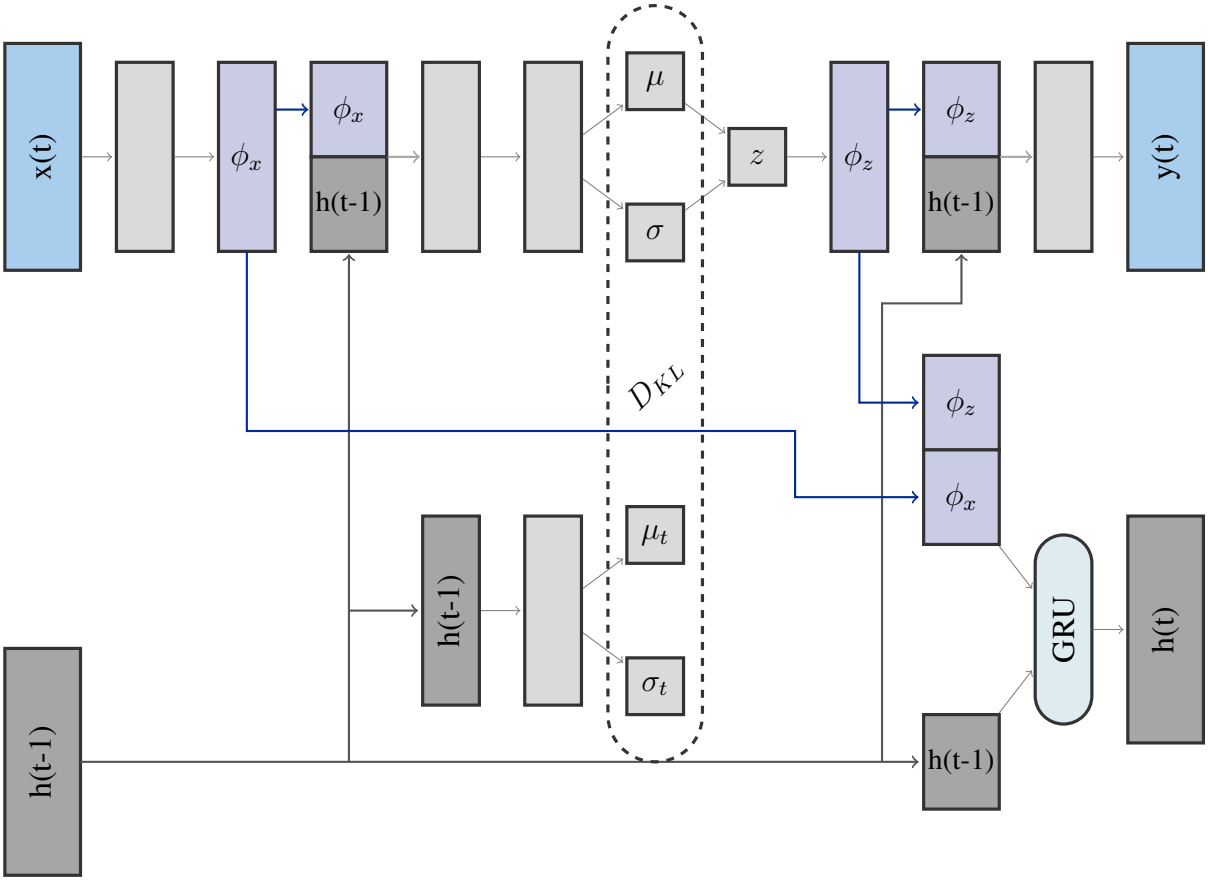

Figure 3: A Variational Recurrent Neural Network cell.

The VRNN used in this study has the architecture shown in Figure 3. The number of neurons in each intermediate layer, including the hidden state and feature extracting layers, but not including the latent space and its μ and σ layers, is 16. The latent space is chosen to be two-dimensional. Since four-vector constituents of jets are being modeled, the input $x(t)$ and output $y(t)$ layers are three dimensional, corresponding to the p_T, η,and ϕ of each constituent. ReLU activations are used in each layer of the network, except for σ and σ_t, which have softmax activations, and z and $y(t)$, which have linear activations.

The constituents of an input jet are processed sequentially, one per each time-step. Each time-

step contributes a loss based on the VAE loss function:

$$\mathcal{L}(t) = MSE + \lambda D_{KL} \tag{12}$$

λ is a factor which weights the KL-Divergence contribution relative to the MSE reconstruction loss. Since harder constituents contribute more information toward the identification of jet substructure, λ is defined to be be a function of constituent p_T fraction relative to the jet's total p_T. Furthermore, since the distribution of constituent p_T fractions depends directly on the number of constituents in a jet, the constituent p_T fraction distribution for each jet is averaged over the entire input dataset to avoid correlations with constituent multiplicity.

In this result, the loss is therefore computed for each constituent as:

$$\mathcal{L}(t) = MSE + 0.1\overline{p_T}(t)D_{KL} \tag{13}$$

Here, MSE is the mean-squared-error between $x(t)$ and $y(t)$, D_{KL} is the KL-Divergence from the current time-step's prior distribution and the encoded posterior, and $\overline{p_T}(t)$ is the p_T of the dataset-averaged constituent? jet at time-step t. The final loss is computed by averaging the individual time-step losses over the entire jet:

$$\mathcal{L} = \frac{\Sigma\mathcal{L}(t)}{N} \tag{14}$$

The hyperparameters involved in this implementation, namely the dimensionality of intermediate layers, and the additional weight coefficient of 0.1 in the loss function were determined via a hyperparameter optimization scan.

After the network is trained, an *Anomaly Score* can be determined for each jet. The KL-Divergence term has been shown to provide better discrimination between anomalous and standard jets than either the reconstruction error or the loss term as a whole.JG: motivate that? Therefore, the Anomaly Score is defined in terms of the KL-Divergence of each constituent, averaged over the whole jet, and restricted to the range of (0, 1) via exponentiation.

$$\text{Anomaly Score} = 1 - e^{-\overline{D_{KL}}} \tag{15}$$

3 Data Samples and Pre-Processing

The performance of the model is investigated by studying its ability to discriminate signal from background in a contaminated dataset of background QCD dijet events with varying amounts of signal. The signal events are a process of the form of $Z' \to XY \to JJ$ where X and Y are two heavy resonances each decaying into a boosted jet J. Independent signal events were generated with the X and Y objects both decaying to either two or three-pronged substructure. The masses of the particles in the signal hypothesis are 3.5 TeV, 500 GeV, and 100 GeV for Z', X, and Y respectively. A total of 99457 signal events were generated for each substructure hypothesis, along with 995,453 background events. The events were generated using PYTHIA8 and DELPHES 3.4.1 with no pileup or MPI included, and selected using a single large-radius (R=1) jet trigger with a p_T threshold of 1.2 TeV [13]. The dataset was provided as part of the LHC Olympics challenge for the ML4Jets2020 Workshopreference.

JG: introduce 3 prong signals

The data is provided as a list of hadrons for each event. The hadrons were then clustered into jets using the anti-k_t jet-clustering algorithm with a radius parameter of 1.0 [3]. To test the

model's performance with varying amounts of contamination, datasets were produced with 10 signal event fractions in the range of 0.01% to 10.0% along a logarithmic scale. To retain as many similarities as possible between tests at different contamination levels, the same set of background events are used for each test while only the amount of signal is varied to match the desired contamination. This corresponds to a total of 895113 background events to accommodate the highest contamination level of 10%.

One of the most consequential elements of this study is the choice of pre-processing. Since the goal is to identify jets mainly due to their substructure, it is important that the model's Anomaly Score does not correlate with other jet features, namely mass and p_T. A common practice to avoid such a correlation in neural network jet modeling architectures is the use of adversarial decorrelation networks (REFS). Applying such adversarial architectures to a VRNN is a complex task which is outside of the scope of this study [17].

3.1 Boosting

The primary goal of pre-processing in this study is to ensure that each jet input into the VRNN is transformed in a way which removes information about differences in mass and p_T so that the VRNN is unable to directly learn and correlate its anomaly score with these features. The resulting pre-processed jets should therefore be superficially identical, with the only differences appearing in the arrangement of their constituents due do varying substructure. This procedure is inspired by a study based on jet images, where a pre-processing method which boosts each jet to the same reference frame allows for a model trained on the pre-processed jets to be robust against variations in mass and p_T [18]. The process can be briefly summarized in three steps:

- Rescale each jet to the same mass
- Boost each jet to the same energy
- Rotate each jet to the same orientation

Algorithm 1 describes in detail the implementation of the rescaling, boosting, and rotating processes, or simply *boosting* for short.

Algorithm 1: Jet Boosting

while *Number of constituents* > 20 **do**
  while *At least one constituent outside of* $\Delta R = 1$ *from jet axis* **do**
    Boost jet in z direction until $\eta_{Jet} = 0$
    Rotate jet about z axis until $\phi_{Jet} = 0$
    Rescale jet mass to 0.25GeV
    Boost jet along its axis until E_{Jet} =1GeV
    Rotate jet about x axis until hardest constituent has $\eta_1 = 0, \phi_1 > 0$
    if *Any constituents have* $\Delta R > 1$ **then**
      Remove all constituents with $\Delta R > 1$
      Rebuild jet with remaining constituents
    else
      break
    end
  end
  if *Number of constituents* > 20 **then**
    Keep up-to the first 20 constituents, ordered in p_T
    Rebuild jet with remaining constituents
  else
    break
  end
end
Reflect constituents about ϕ axis such that the second hardest constituent has $\eta_2 > 0$

To evaluate the efficacy of this procedure, the model is trained on a dataset with 10% signal contamination both before and after pre-processing, and the resulting correlation between Anomaly Score and jet mass is compared. Figure 4 shows the two-dimensional distribution of leading jet mass vs. anomaly score before and after boosting the input jets. The results depict a significantly smaller amount of correlation between the jet's mass and its Anomaly Score, as desired.

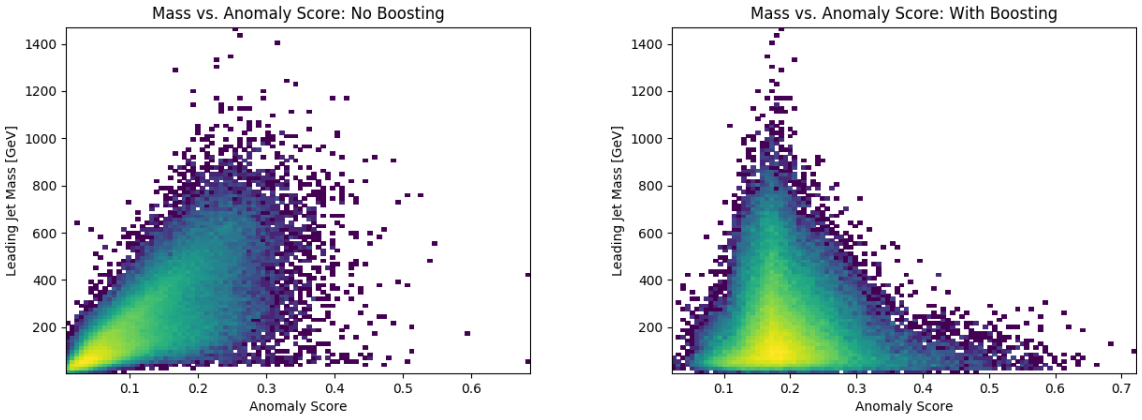

Figure 4: Leading Jet Mass vs Anomaly Score distributions before (left) and after (right) applying our boosting method

3.2 Sequence Ordering

In addition to this boosting method, the effect of *sequence ordering* on the input constituents has been investigated. In fixed architecture models, such as VAEs or image based Convolutional Neural Networks (CNNs), the ordering of constituents in the list of training inputs is seldom important. However, in recurrent architectures such as the VRNN, choosing a clever sequence ordering method that highlights important sequence features can boost performance.

The most intuitive method of ordering jet constituents is by their p_T in decreasing order, as harder constituents contribute more to a jet's substructure than softer constituents. The objective of this study is to build a model which can differentiate between diffuse jets resulting from soft QCD interactions, and jets with multiple cores resulting from the hadronic decay of boosted objects. Therefore, it is favorable to use a sequence ordering which makes the existence of multiple hard cores of a jet distinctly apparent. This is achieved by ordering the constituents in k_t-distance order. More specifically, the n^{th} constituent in the list is determined to be the constituent with the highest k_t-distance relative to the previous constituent, with the first constituent in the list being the highest p_T constituent.

$$c_n = max(p_{Tn}\Delta R_{n,n-1}) \tag{16}$$

The effect on performance due to this choice of constituent ordering can be easily illustrated in the case of a two-prong jet. In such a case, the sequence will start with a constituent in one of the two cores of the jet, and be subsequently and consistently followed by a constituent belonging to the other core. This results in an easily predictable pattern which the VRNN is more able to model, particularly compared to a homogenous QCD jet. The resulting performance difference between p_T sorted and k_t sorted inputs is shown in Figure 5. Using the same 10% contaminated dataset, two-prong signal jets have a notably lower anomaly score when compared to background QCD jets due to the ease of modeling their substructure. (diagram maybe?)

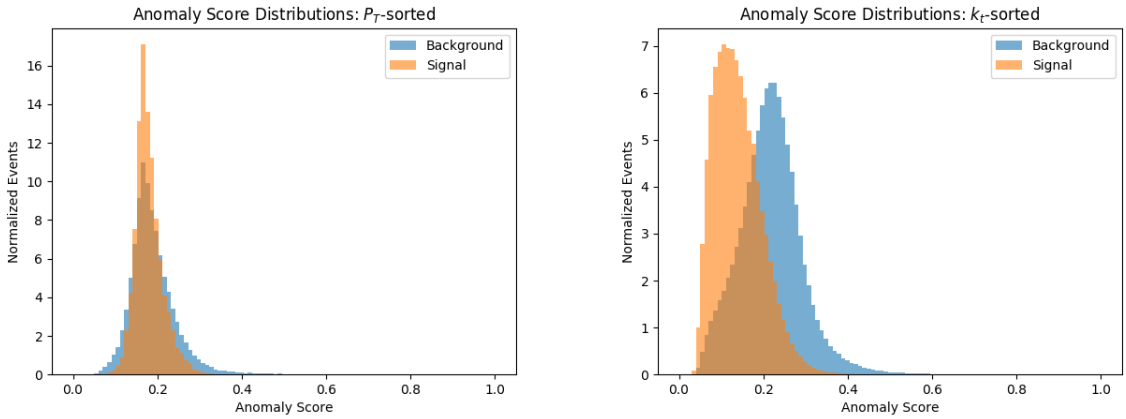

Figure 5: Leading jet Anomaly Score distributions for background and signal events, before (left) and after (right) applying k_t-sorted ordering on constituents for training input jets.

4 Results

JG: this paragraph still feels awkward to me Since the model is designed to discriminate between signal and background at the jet level, its basic performance can be studied by using only the leading jet of each event. In addition, since a full description of each event is available in the generated dataset, the score can be applied to jets and used to discriminate between signal and background in an event-level analysis context. The Anomaly Score discrimination performance is thus given for distinguishing between signal and background jets, and as the sole event discriminator in a search for the Z' particle.

4.1 Jet Level Performance

In the jet-level assessment, the model is trained on the leading jets of each event. To evaluate the trend in performance during training, a computation of the Receiver Operating Characteristic's Area Under the Curve (ROC AUC) is performed after each epoch by comparing events in either the training set or the background-only validation set to those in the pure signal set. Figure 6 shows the results of this training scenario in the case of 10% contamination. Notably, the model quickly reaches optimal performance, and retains a stable performance throughout the training period. Also shown is the same trend on an independent validation set of data, which is comprised of background-only events, and shows similar stability during training. It is important to note that the feature of the ROC AUC being lower than 0.5 is expected, and is a result of the k_t-ordered sequencing. JG: I don't understand this. which allows the model to reconstruct jets with two-prong substructure more easily than soft QCD-like jets with more diffuse substructure.

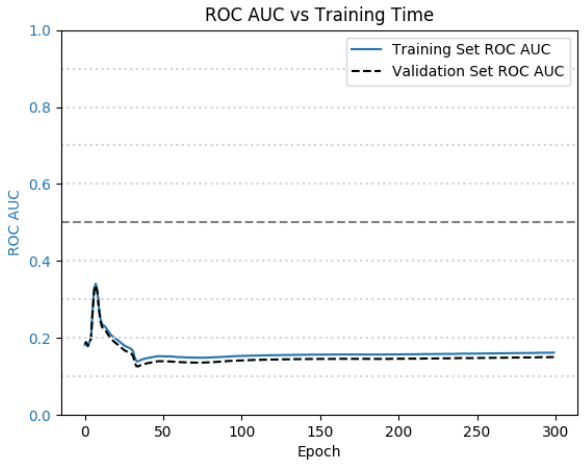

Figure 6: Area Under the Curve (ROC AUC) vs. training time in epochs. Due to the nature of the k_t-ordered sequencing, values closer to zero represent higher performance in separating signal jets from background jets. (NOTE:Waiting on new plot

Since the training scenario is entirely unsupervised, the resulting Anomaly Score distributions from each training dataset may vary. To arrive at a consistent score distribution, a transformation is applied on the resulting Anomaly Score which aims to satisfy two conditions:

- The mean of the resulting distribution is at an anomaly score value of 0.5.
- Anomaly scores closer to a value of 1 correspond to more signal-like jets.

The transformation can be summarized as

$$\rho' = 1 - \left(\frac{\rho}{2\overline{\rho}}\right) \tag{17}$$

where ρ' is the transformed Anomaly Score, and $\overline{\rho}$ is the mean of the un-transformed Anomaly Score distribution of the training set.

JG: I think we need to explain what datasets this is evaluated in. After applying this transformation, an optimal cut on the Anomaly Score is determined by comparing the signal to background ratio at each potential value, as shown in Figure 7. The resulting peak signal to background corresponds to an optimal cut on the transformed Anomaly Score at a value of 0.75.

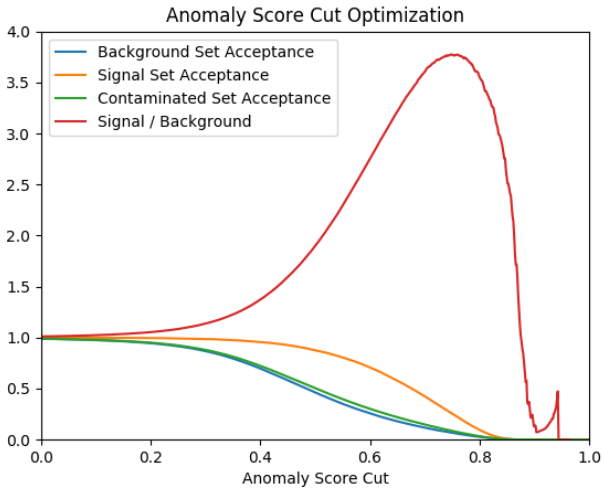

Figure 7: Anomaly Score cut value optimization, where the peak signal to background ratio corresponds to the analysis cut value of 0.75.

Figure 8 shows the mass distributions of the leading jet before and after a jet-level selection requiring the Anomaly Score to exceed a value of 0.75. The dataset to which this cut is applied is composed of 10% two-prong signal events and 90% background. Notably, the presence of the known resonances at 100 GeV and 500 GeV is enhanced. Sculpting in the background distribution is observed, which is an effect of mass correlation mainly introduced by the k_t-ordered sequencing. Both the signal enhancement and background sculpting are similarly observed on three-pronged signatures in Figure 9, also shown in a 10% contaminated dataset.

JG: its sort of tough to convince yourself that the post-cut bump is more prominent... would an S/B in that mass region be helpful to quantitatively describe?

Fig 8 and 9 title should not say dijet mass

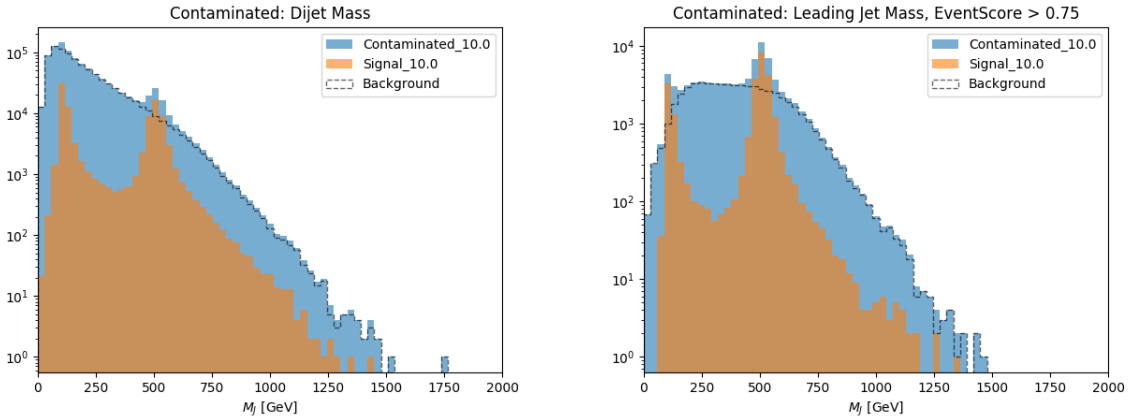

Figure 8: Leading jet mass distributions with a two-prong signal hypothesis before (left) and after (right) a cut on the Anomaly Score

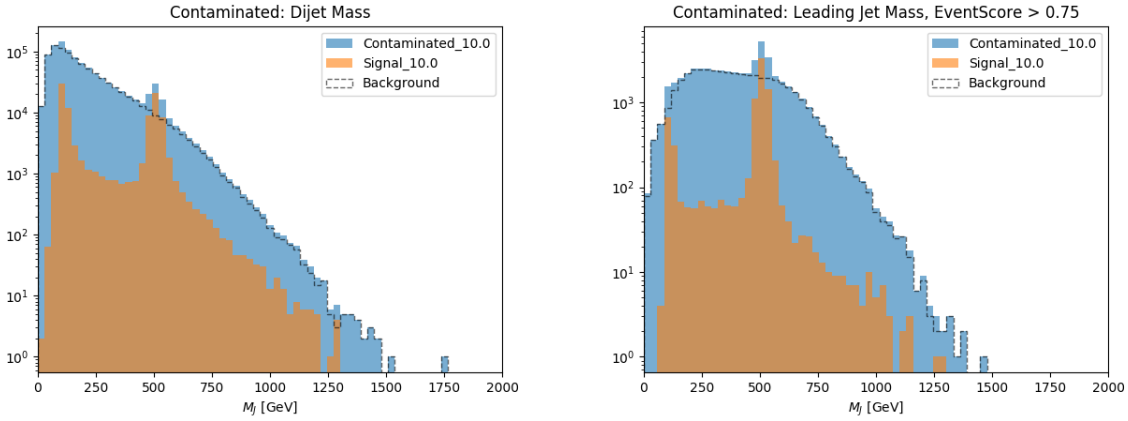

Figure 9: Leading jet mass distributions with a three-prong signal hypothesis before (left) and after (right) a cut on the Anomaly Score

As the Anomaly Score in this context distinguishes two-pronged substructure from homogenous jets, it is apt to compare to a commonly used high-level variable sensitive to two-pronged signals. The energy correlation function ratio D_2 is selected and used a benchmark to contextualize the Anomaly Score in both signal discrimination and jet mass correlation.

Figure 10 shows a comparison of the shapes of the contaminated jet mass distributions when subject to a cut on Anomaly Score and D_2. The Anomaly Score cut is at the optimized value of 0.75, and the D_2 cut is chosen to give the same acceptance. The shape of the jet mass distribution is more significantly sculpted after the D_2 selection than the Anomaly Score selection, indicating more significant correlation of D_2 with jet mass. Such a result is expected, given that the Anomaly Score is determined only from jet constituent four-vector information, without any high-level information being input into the model.

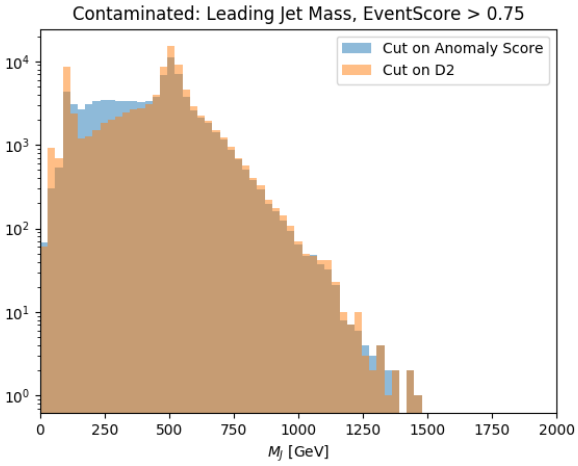

Figure 10: Contaminated set shape comparison between a selection on the Anomaly Score and a selection on the D_2 variable with equal acceptance.

Another important study involves the model's performance over a range of signal contamination levels. Figure 11 shows the ROC AUC values of both two and three-pronged signal hypotheses after training on each of the contaminated sets. Notably, the performance is consistent along all contaminations, and effective on two different prong multiplicities without any prior substructure hypothesis. The Anomaly Score can thus be interpreted as a quantity which is capable of adequately and consistently parametrizing multiple distinct substructure scenarios. This feature is valuable in model-independent searches, or those without a pre-defined signal substructure hypothesis.

The ability of the Anomaly Score to be consistently performant along a large range of contaminations is unexpected in the context of anomaly detection, where the dilution of the training set with a high number of signal elements results in lower performance. Here, the consistent performance can be attributed to the choice of k_t-ordered sequencing and the representation of jets as variable-length sequences of constituents. JG: I don't think we can just say that these are the reasons... without some further proof.

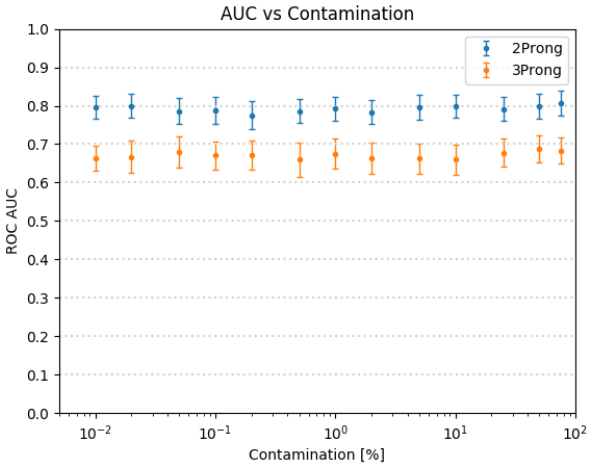

Figure 11: ROC AUC vs. signal contamination in training dataset, given as a percentage of the total events.

4.2 Event Level Performance

A natural extension of the Anomaly Score's ability to distinguish anomalous jets is to apply the score in an analysis-like context. In this study, the goal is to reconstruct the Z' particle in the invariant mass spectrum M_{JJ} of the two jets in each signal event. To do this, the network is trained on both the leading and sub-leading jets of the event, with one set of network weights saved for each amount of contaminationJG: explain what contaminations you train on?.

Since the model produces one Anomaly Score per jet, a combination of Anomaly Scores for the leading and sub-leading jet must be combined to arrive at an overall *Event Score*. In this study, the Event Score is chosen to be the highest of the two individual Anomaly Scores between the leading and sub-leading jets. This constructs an event-level discriminant which uses the most anomalous jet in the event to discriminate. The ability of the Event Score to distinguish signal from background is illustrated in Figure 12, showing the correlations between the dijet invariant mass and the assigned Event Score in a contaminated dataset. The significant feature of the 3500 GeV Z' occupies high values of the Event Score, validating the Event Score as a discriminant of anomalous events from background.

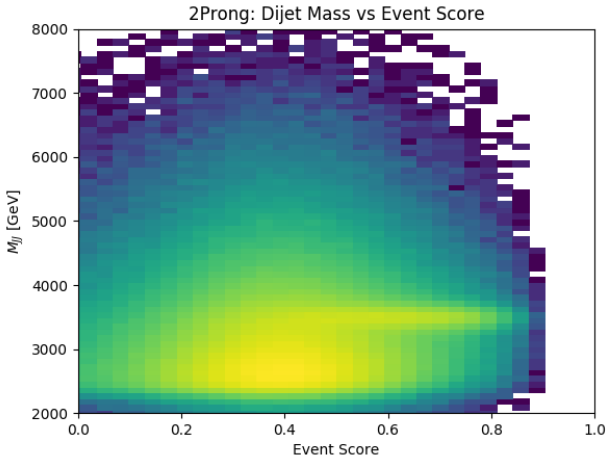

Figure 12: Dijet invariant mass vs. Event Score.

The goal of this search is to observe an excess of signal events in the dijet invariant mass distribution at a mass corresponding to the Z' particle. This is commonly referred to as a "bump hunt" search, in which the signal is expected to appear as a bump upon an otherwise smoothly falling background distribution.

Figure 13 shows the dijet mass distributions of the signal, background, and contaminated datasets, before and after applying a selection on the Event Score at a value of 0.65. This cut value was chosen such that the performance of the anomaly score is demonstrated while still retaining enough statistics to preserve the shape of the falling background distribution. Also plotted is the local significance σ in each bin of the corresponding histogram, where a total uncertainty of 5% on the number of background events is assumed. The local significance was computed using the BINOMALEXPZ function from ROOSTATS [16].

The requirement on the Event Score dramatically increases the significance of the excess from 1σ to 6σ at a signal contamination of 0.5%, while still retaining the smoothly falling behavior of the background. No selections other than the Event Score requirement have been applied in these scenarios besides the initial trigger requirement of $p_T > 1200GeV$ on the leading jet. In Figure 14, a similar result is seen in the case of the three-pronged signal, where the Event Score requirement results in an increase of local significance of the signal peak from 2σ to 4σ at a signal contamination of 1% under the same conditions. JG: maybe name these signals specifically, what even is the 3 prong signal, is it Z'? These results display the capability of the Anomaly Score as an analysis variable, as it can distinguish signal events while being robust against ambiguities in signal jet mass, p_T, and substructure.

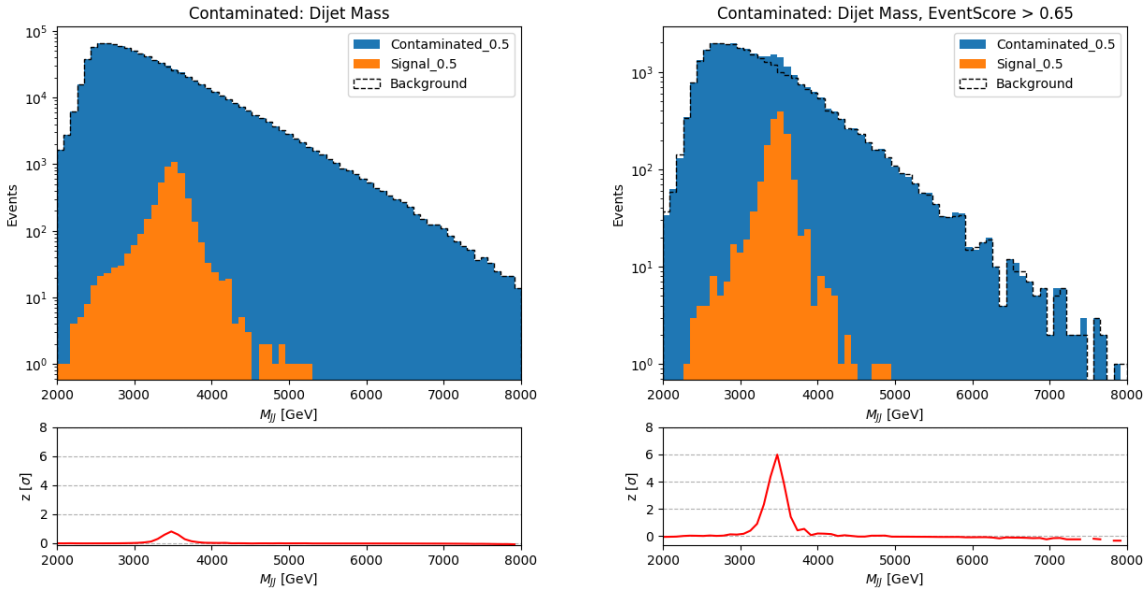

Figure 13: Two-prong dijet mass distributions before (left) and after (right) a cut on the Event Score, at a signal contamination of 0.5%.

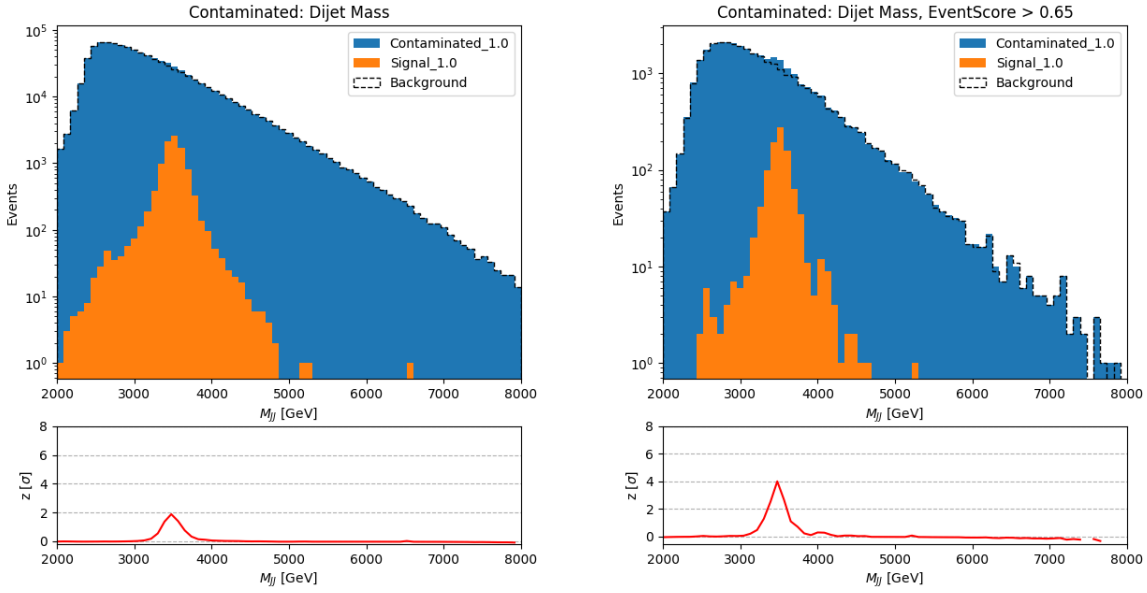

Figure 14: Three-prong dijet mass distributions before (left) and after (right) a cut on the Event Score, at a signal contamination of 1.0%.

Conclusion

A novel approach for anomaly detection in the context of new physics searches is presented. The technique utilizes a Variational Recurrent Neural Network trained on contaminated datasets to

distinguish jets resulting from boosted hadronically decaying objects from those resulting from soft QCD processes. A precise pre-processing procedure is developed, in which each input jet is boosted to the same reference mass, energy, and orientation, coupled with a sequence ordering which makes the presence of signal-like substructure more apparent. The resulting training produces an Anomaly Score per jet that is sensitive to multiple substructure hypotheses. In a jet-level study that only uses leading jets in the dataset, the model enhances signal with less jet mass correlation than other traditional substructure variables such as D_2. In addition, the resulting Anomaly Score is equally performant across varying levels of contamination, allowing for a consistent characterization of substructure regardless of the amount of signal present in the dataset. When applied to an event-level context, the Anomaly Score greatly increases the significance of signal excesses regardless of substructure hypothesis, while retaining the smoothly falling shape of the background's mass distribution.

The Variational Recurrent Neural Network used in this study is a powerful tool capable of learning underlying features of physics objects presented as sequential data. Its applications to new physics searches are numerous, with one of the most attractive features being the potential for training directly on data without a pre-defined signal substructure hypothesis. It is also a general tool for modeling sequential data of any type, making it compatible with common high energy physics tasks such as event-level searches or known object tagging and identification.

The approach in this study is somewhat model-independent, in that it accommodates multiple substructure hypotheses. However, the model lends itself to a number of potential avenues for exploration into traditional supervised contexts as well, expanding its utility beyond the context of anomaly detection. Since the overall structure of the model contains both elements of Variational Autoencoders and Recurrent Neural Networks, more complicated architectural iterations can be employed as natural extensions of the VRNN. Examples of possible additions include adversarial mass de-correlation networks, and conditional architectures which can supplement the VRNN's input by a fixed length vector of high-level jet features. Further study into the ordering of the input constituent sequence is also warranted, as this feature may be able to be tuned to accommodate more precisely defined analysis contexts or substructure hypotheses.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. PHY-2013070.

References and Notes

[1] J. An and S. Cho. Variational autoencoder based anomaly detection using reconstruction probability. 2015.

[2] D. Bank, N. Koenigstein, and R. Giryes. Autoencoders, 2020.

[3] M. Cacciari, G. P. Salam, and G. Soyez. The anti-kt jet clustering algorithm. *Journal of High Energy Physics*, 2008(04):063–063, Apr 2008.

[4] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

[5] J. Chung, K. Kastner, L. Dinh, K. Goel, A. Courville, and Y. Bengio. A recurrent latent variable model for sequential data, 2016.

[6] A. Collaboration. Observation of a new particle in the search for the standard model higgs boson with the atlas detector at the lhc. *Physics Letters B*, 716(1):1–29, Sep 2012.

[7] C. Collaboration. Observation of a new boson at a mass of 125 gev with the cms experiment at the lhc. *Physics Letters B*, 716(1):30–61, Sep 2012.

[8] M. Farina, Y. Nakai, and D. Shih. Searching for new physics with deep autoencoders. *Physical Review D*, 101(7), Apr 2020.

[9] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[10] P. D. Group. Review of Particle Physics. *Progress of Theoretical and Experimental Physics*, 2020(8), 08 2020. 083C01.

[11] T. Heimel, G. Kasieczka, T. Plehn, and J. Thompson. Qcd or what? *SciPost Physics*, 6(3), Mar 2019.

[12] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, Nov. 1997.

[13] G. Kasieczka, B. Nachman, and D. Shih. R&D Dataset for LHC Olympics 2020 Anomaly Detection Challenge, Apr. 2019.

[14] D. P. Kingma and M. Welling. Auto-encoding variational bayes, 2014.

[15] E. M. Metodiev, B. Nachman, and J. Thaler. Classification without labels: learning from mixed samples in high energy physics. *Journal of High Energy Physics*, 2017(10), Oct 2017.

[16] L. Moneta, K. Belasco, K. Cranmer, S. Kreiss, A. Lazzaro, D. Piparo, G. Schott, W. Verkerke, and M. Wolf. The roostats project, 2011.

[17] S. Purushotham, W. Carvalho, T. Nilanon, and Y. Liu. Variational recurrent adversarial deep domain adaptation. In *ICLR*, 2017.

[18] T. S. Roy and A. H. Vijay. A robust anomaly finder based on autoencoders, 2020.