The same de-correlation technique also allows us to tackle the third challenge. A promising approach in this direction is to combine well-understood features like mass peaks with implicit, orthogonal information [27]. More generally, we will show how any well-defined physics effect can be de-correlated from the autoencoder analysis, allowing us to construct control regions, side bands, or a flat spectrum suitable for a bump hunt in any observable needed for a given analysis.

In Sec.2.1 we will start by constructing an autoencoder [15–17] based on a convolutional network [28], in our case the image-based DeepTop tagger [11,12]. Alternatively, we can analyze 4-vectors in an autoencoder version of the DeepTopLoLa tagger, as shown in Sec. 2.2. Next, we will control what kind of information the network uses by taking out the jet mass distribution through an adversarial network in Sec. 2.3. This allows us to devise a convincing side band analysis on the jet mass for the anomaly search [27]. With the help of these sidebands we can study the stability of the autoencoder network trained on not fully controlled, impure QCD samples in Sec. 2.4. After establishing our new methods using top tagging we will test them on scalar decays to four jets in Sec. 3.1 and on non-QCD showers in Sec. 3.2.

2 Autoencoded QCD vs tops

The aim of our study is to identify jets with an exotic, non-QCD origin using a neural network that is only trained on QCD jets. This can be done with autoencoder networks, which are stacks of networks layers with an intermediate set of bottleneck layers with a strongly reduced number of units, corresponding to a latent space with reduced dimensionality. Such a bottleneck can be added to convolutional networks [28], but it can also be added to a LoLa-like network working on constituent 4-vectors. The main structural change is that autoencoders do not work towards an output value which, assuming the right loss function, gives a probability for a jet being either signal or background. Instead, the network on both sides of the bottleneck is approximately symmetric, and the loss function is the difference between the input and the output. Once we run such a trained network on a test sample the loss function will tell us how well the network with its bottleneck describes the features of the test sample.

is this the right word? or preserves? encodes?

An established, albeit non-glamorous benchmark for subjet studies are boosted hadronic top decays. This is why we first test our new autoencoder setup, trained on QCD jets, for anomalous top jets. After we benchmark autoencoders for image-based and 4-vector-based architectures, we will introduce a de-correlation with the jet mass. This approach can be immediately generalized to any other variable, defining plenty of control regions and side bands to control the autoencoder in an experimental reality.

2.1 Jet images

see above… pixellated energy not calorimeter?

As long as we focus on the calorimeter information, we can analyze jets using a CNN to learn jet images. Our autoencoder architecture is based on the DeepTop tagger [11], with significant improvements especially to the image pre-processing, developed in Ref. [12].

Our top and QCD samples are similar to the sample used in our DeepTop studies [11,13]. We simulate top pair and di-jet production with Pythia8.2.15 [29] and Delphes3.3.2 [30] for a collider energy of 14 TeV. For the QCD sample we do not distinguish between hard

I don't trust DELPHES to faithfully describe ATLAS & CMS. What does it actually bring here? I hope the pixellation of the energy into your grid is the dominant effect. Can you test this?

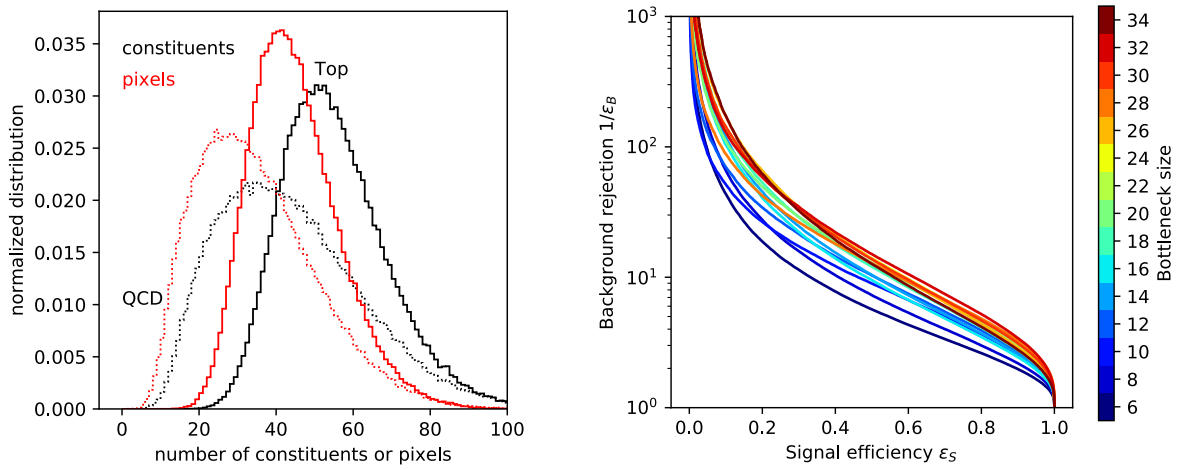

Figure 1: Left: numbers of constituents and of non-zero pixels for tops and QCD jets, 400,000 jets in total. Right: ROC curves for the image-based autoencoder identifying anomalous top jets for different bottleneck sizes.

Most of these "specialised tools" would these days use groomed jets. Do your networks still work on groomed jets? Can you show this?

quarks and gluons. We ignore multi-parton interaction and pile-up, assuming that there are specialized tools available to remove it [31]. The substructure containers are fat anti-k_T jets [32] with distance parameter $R = 0.8$, defined by FastJet3.1.3 [33]. They are required to have a transverse momentum in the range

$$p_{T,j} = 550 \ldots 650 \text{ GeV} . \tag{1}$$

In addition they must be central, $|\eta_j| < 2$. For all signal jets we require a truth-level partonic top to be within the area of the fat jet. The objects of the subjet analysis are particle flow objects [34] from the Delphes E-flow. In the left panel of Fig. 1 we show the number of particle flow constituents for signal and background jets. The main feature is that already based on the larger number of constituents we could identify the hadronic top decays.

Following Ref. [12] we employ an improved pre-processing of the jet images, most notably applied before pixelization. This approach is directly motivated by the particle flow approach, which combines the coarse calorimeter information with the high-resolution tracker and provides us with a set of high-resolution 4-vectors. The center of the image is not defined by the hardest object, but by the k_T-weighted centroid of the fat jet constituents. The major principle axis is then turned to 12 o'clock. Finally, the image is flipped along the x-axis and y-axis, to ensure that the hardest constituent is located in the upper right quadrant. Only after this pre-processing we pixelize the images into a 40×40-pixel image, covering $\eta = -0.57 \ldots 0.57$ and $\phi = -0.69 \ldots 0.69$ around the center of the fat jet. The entries of the calorimeter images are given by the transverse momentum entering the detector cell, *i.e.* the sum of the transverse momenta of all particle flow objects covered by a pixel. Also in the left panel of Fig. 1 we show the number of non-zero pixels per image. The full image with its 1600 pixels is indeed sparsely filled. Each of the pixels is finally normalized to the sum of all pixels in the jet image. These images define the input and the output format of the autoencoder network.

The architecture of the autoencoder network is shown in Fig. 2. We use Keras [35] combined with Tensorflow [36] to build our network. It is almost symmetric between the

input and the output. The loss function is simply

$$L_{\text{auto}} = \sum_{1600 \text{ pixels}} \left(k_T^{\text{norm,in}} - k_T^{\text{auto}} \right)^2 , \tag{2}$$

in terms of the normalized input image and the autoencoder output image. We use the PReLU activation function throughout the network, to avoid a zero pseudo-solution, except for a linear activation function in the last layer. We use the Adam optimizer [37] for the learning rate.

The autoencoder is trained on 100,000 QCD or background jets for up to 100 epochs and allow for an early stopping after ten epochs with stable loss. Our test sample consists of 200,000 top jets and 200,000 QCD jets. The large test sample allows for a study of the performance on several independent samples, confirming that our ROC curves are stable. For a variable cut in the loss function we can evaluate the composition of the signal-like jets in terms of true top and true QCD jets. These two fractions define a ROC curve, as shown in the right panel of Fig. 1. For these curves we vary the size of the bottleneck from 6 to 34 units in the smallest dense layer shown in Fig 2. We see a sizeable variation with the bottleneck size, developing a stable high-performance plateau between 20 and 34. It gives a stable area under curve (AUC) around 0.89 with a loss around 10^{-5} per pixel. The size of the bottleneck has to be compared with the initially 1600 pixels, of which 10 to 70 are non-zero, and which the CNN pools to 400 combined pixels. This large bottleneck size indicate that the image architecture is not perfectly adapted to encode the relevant QCD vs tops information in a small network layer.

The large size of the test sample allows us to evaluate our autoencoder on separate, statically independent test samples. While the corresponding spread does not account for systematics uncertainties related to the training, especially the training on data, it defines a statistical uncertainty of the autoencoder. It is shown as widths of the ROC curves, which are generated by evaluating the network on ten independent test samples with 20,000 QCD jets and 20,000 top jets each.

2.2 LoLa

When we want to include information beyond the calorimeter output, we can for example use the neural network based on the constituent 4-vectors developed for the DeepTopLoLa

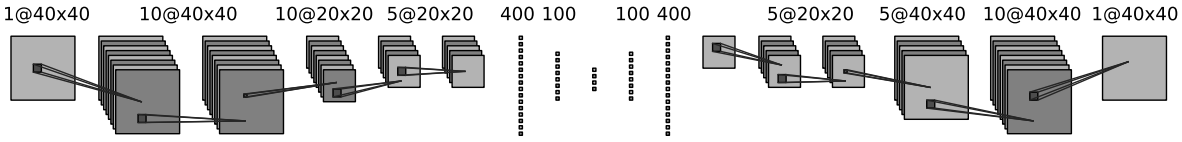

Figure 2: Architecture of the image-based autoencoder network. The 40×40 images are average-pooled to 20×20 images before entering the bottleneck. The dense units are first reduced from 400 to 100, the minimum size at the bottleneck is variable.

tagger [13]. It starts from a set of measured 4-vectors sorted by transverse momentum

$$(k_{\mu,i}) = \begin{pmatrix} k_{0,1} & k_{0,2} & \cdots & k_{0,N} \\ k_{1,1} & k_{1,2} & \cdots & k_{1,N} \\ k_{2,1} & k_{2,2} & \cdots & k_{2,N} \\ k_{3,1} & k_{3,2} & \cdots & k_{3,N} \end{pmatrix} . \tag{3}$$

Following the left panel of Fig. 1 we use $N = 40$ constituents, after checking that an increase to $N = 120$ does not make a measurable difference. For jets with fewer constituents we naturally fill the entries remaining in the soft regime with zeros.

What do you use for constituent masses? Zero? Truth particle mass? Or?

To remove all information from the jet-level kinematics we boost all 4-momenta into the rest frame of the fat jet. This also improves the performance of our network. Inspired by recombination jet algorithms we can add linear combinations of these 4-vectors with a trainable matrix C_{ij}, defining a combination layer

$$k_{\mu,i} \xrightarrow{\text{CoLa}} \widetilde{k}_{\mu,j} = k_{\mu,i}\, C_{ij} \qquad \text{with} \quad C = \begin{pmatrix} 1 & 1 & 0 & \cdots & 0 & C_{1,N+1} & \cdots & C_{1,M} \\ \vdots & 0 & 1 & & \vdots & C_{2,N+1} & \cdots & C_{2,M} \\ \vdots & \vdots & \vdots & \ddots & 0 & \vdots & & \vdots \\ 1 & 0 & 0 & \cdots & 1 & C_{N,N+1} & \cdots & C_{N,M} \end{pmatrix} . \tag{4}$$

We allow for $M = 10$ trainable linear combinations. These combined 4-vectors carry information on the hadronically decaying massive particles. In the original LoLa approach we map the momenta $\tilde{k}_j$ onto observable Lorentz scalars and related observables [13]. Because this mapping is not easily invertible we do not use it for the autoencoder. Instead, we extend the 4-vectors by another component containing the invariant mass,

Why is this needed when there is no independent info added just formatting? Or does the network just not understand what 4-vectors are?

$$\tilde{k}_j = \begin{pmatrix} \tilde{k}_{0,j} \\ \tilde{k}_{1,j} \\ \tilde{k}_{2,j} \\ \tilde{k}_{3,j} \end{pmatrix} \xrightarrow{\text{LoLa}} \begin{pmatrix} \tilde{k}_{0,j} \\ \tilde{k}_{1,j} \\ \tilde{k}_{2,j} \\ \tilde{k}_{3,j} \\ \sqrt{\tilde{k}_j^2} \end{pmatrix} . \tag{5}$$

This defines a set of 51 extended 4-vectors, which form the input to our neural network. Again, we use Keras [35] combined with Tensorflow [36]. Its architecture is shown in Fig. 3. The layer immediately after the LoLa contains $51 \times (4 + 1) = 255$ units. Between the second layer after LoLa and the last layer, the autoencoder network is symmetric. The final output consist of 40 4-vector-like objects, which can be compared with the corresponding

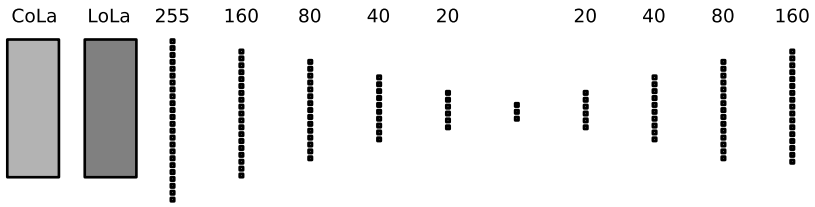

Figure 3: Architecture of the 4-vector-based autoencoder network. The 255 input units correspond to 55 LoLa-vectors with $4+1$ entries each. The output only consists of 160 units, because the extended 4-vectors only carry four independent observables.

other standard assumption. On the other hand, the more weakly the question to the data is defined, the more important it is to control what the neural network actually learns. This is especially true when we use the network on low-level information rather then established high-level kinematic observables [38].

An established way to test a network is to exclude known, well-defined pieces of information from it through adversarial networks [23–26]. They consist of two networks playing against each other. Similar to generative adversarial networks, they can be used to train a network as an equivalent replacement for another data generator. In our application the additional adversary is trained to extract for instance the jet mass from the autoencoder output described in Eq.(2). In this image-based case a naive adversary loss function would read

$$L_{\text{adv}}(M) = \left[\widetilde{M}\left(\left|k_{T,i}^{\text{adv}} - k_{T,i}^{\text{auto}}\right|\right) - M\right]^2 , \tag{7}$$

with the inputs $k_{T,i}^{\text{auto}}$, the outputs $k_{T,i}^{\text{adv}}$, the given jet mass M, and the trained proxy to the jet mass $\widetilde{M}$. As we will discuss below, for our study we replace the exact function $\widetilde{M}$ with a binned determination of the jet mass [24]. The combined loss function which replaces Eq.(2) for the autoencoder can be written in terms of a Lagrangian multiplier [23, 24]

$$L = L_{\text{auto}} - \lambda\, L_{\text{adv}}(M) . \tag{8}$$

The Lagrangian multiplier λ introduces a boundary condition, $L_{\text{adv}} \to 0$, in case the adversary learns the mass perfectly. The value of λ determines the balance between the two networks. While the task of the autoencoder network is to describe the QCD training data, the adversary extracts the jet mass from the autoencoder output. Playing against each other and minimizing the combined loss function with the relative sign, the combined network wants the adversary to be as unsuccessful as possible. The adversarial autoencoder will hence avoid all information on the jet mass or any other boundary condition. Note that at least for the top jets this only affects the fat jet mass and still leaves us with the W-mass in the clustering history.

As a starting point, we show the jet mass distribution after applying the image-based autoencoder. We know from many studies that the jet mass is the single most powerful observable in separating QCD jets from hadronically decaying heavy states. On the other hand, since we also know that a small fraction of QCD jets will feature large jet masses, we expect to see a top signal as a jet mass peak over a smooth QCD jet background.

In the left panel of Fig. 5 we show jet mass distributions for QCD jets in slices of the autoencoder loss function. The per-centile ranges from all QCD jets to the 5% least QCD-like of all QCD jets. For the full jet sample we see the expected peak at small $m_j \approx 50$ GeV with a long tail extending beyond 300 GeV. For the least QCD-like jets in the pure QCD sample a peak at $m_j \approx 200$ GeV appears. This means that the cut on the autoencoder output badly shapes the background and makes it signal-like. This defines the task of the adversarial network: provide a smooth jet mass distribution for QCD jets, independent of the value of the autoencoder loss function; or in other words, de-correlate the jet mass from the autoencoder.

Again, we use Keras [35] and Tensorflow [36] with the Adam [37] optimizer for the combined adversarial network. The image-based autoencoder part of the network is described in Fig. 2; the adversarial part consists of eight dense layers with 800, 400, 200, 100, 50, 25, 10, and 12 units. We now train this network on 600,000 QCD jets. The output layer corresponds to 10 pre-defined slices in the jet mass, binned such that they are populated by the same

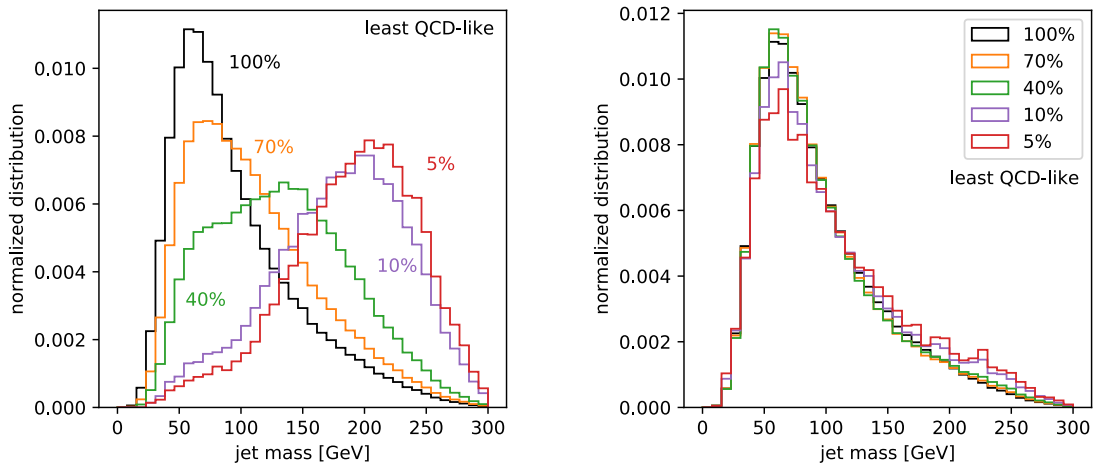

Figure 5: Left: jet mass distributions from the image-based autoencoder applied to QCD jets. The different lines show the full sample up to the 5% least QCD-like jets, defined by the autoencoder loss function. Right: the same jet mass distributions, but for the QCD-trained adversarial autoencoder network.

number of QCD jets. On each side we add overflow bins which are not populated by QCD jets. The task of the adversary is not to extract the exact jet mass value, but to determine the probabilities for the jet mass to fall into each bin. This statistical interpretation requires a multi-label cross entropy as the adversary loss function [24]. All layers use the ReLU activation function except for the last layer, where a SoftMax activation function guarantees that all 12 probabilities sum to one. When training on the combined loss function, each epoch is split into batches of size 128. For each batch we first train the autoencoder using the combined loss function of Eq.(8) and then train the adversary with only the adversary loss function. The size of the Lagrangian multiplier is chosen such that the two contributions to the loss function are of similar size, *i.e.* it balances the de-correlation vs the discrimination power of the network. For instance, the jet mass distribution for $\lambda = 5 \cdot 10^{-4}$, shown in the right panel of Fig. 5, indicates that the background shaping is indeed largely gone.

A little too jargon-heavy?

To study the interplay of the mass de-correlation with the performance of the adversarial autoencoder we show results for three values of λ in Fig. 6. For increasing values of λ the background shaping indeed improves. On the other hand, we can illustrate the performance of the network by testing on QCD data with 3% top jets injected. For the full sample we indeed see a hint of top jets around $m_j = m_t$ in all three panels of Fig. 6. We can then extract the 5% least QCD-like jets, which should include most of the top jets. What we find is that the number of top jets in this selection is diluted from the maximum expected 3/5 of the 5% least QCD-like jets. This dilution grows with λ, because it is an effect of taking out the jet mass as the strongest discriminator from the network. The performance drop is given as AUC values and detailed in the right panel of Fig. 6, where we show the ROC curves for the adversarial autoencoder. As before, we evaluate the network on 10 independent test samples of 20,000 QCD jets and 20,000 top jets.

For the interplay between the mass de-correlation and the performance of the network the ROC curves are not the final word, though. Because the jet mass is removed from the autoencoder, we now see a clear top mass peak in the least QCD-like selection. This peak

training and analysis samples. The key distribution is the jet mass for increasingly anomalous jets. It can be evaluated using standard bump-hunting techniques to extract a new physics signal. The signal jets can then be further dissected using orthogonal analysis techniques.

How would one know if technique was orthogonal? Haven't you in principle used everything except the jet mass already?

Because the training and the search rely on data in the same phase space region, the usually leading systematics do not enter. The remaining key uncertainty is the propensity of the network to induce a fake bump despite adversarial training. It can be reduced through a proper tuning of the hyper-parameters on simulation and verified using additional control regions in data. In case we see no signal, the network response can be used to set exclusion limits for arbitrary signal models. Compared to usual new physics searches the tables are turned: instead of training the network on simulation and applying it to data, we now train the autoencoder on data and apply it to simulation. In turn, the related systematic uncertainties have be considered for exclusion limits.

3 Exotics in jets

While top decay jets a are great tool to test and benchmark our autoencoder, they are clearly not the most attractive application as the top is a known particle. Instead, we need to show how the autoencoder works in extracting other, exotic jets from a QCD sample where the parameters might not a priori be known. We will rely on two examples for this purpose: first we will test the autoencoder on a sample which includes a Higgs-like scalar decaying to four jets. It replaces the second, W-mass handle in the top jet by an increase in the subjet multiplicity. Second, we will use a modified, dark shower with QCD radiation as well as dark radiation off heavy dark quarks. The dark radiation produces missing energy and modifies the jet mass distribution, while leaving two hard jets with anomalous radiation patterns. For both of these models we show how the autoencoder with and without adversary can be used for a signal-independent LHC search.

3.1 Scalar decay to jets

As an alternative to the massive top jets we study a toy model with a Higgs-like scalar decaying to four charm jets through two light pseudoscalars,

$$pp \to (\phi \to aa \to c\bar{c}\, c\bar{c}) + \text{jets} \,. \tag{9}$$

The particle masses are $m_\phi = m_t = 175$ GeV and $m_a = 4$ GeV. We are not concerned with constraints on this toy model and choose the scalar mass such that we can easily compare our results with the top jet case and the pseudoscalar mass such that it decays to, for example, charm jets.

The light pseudoscalar are will be strongly boosted, and its decays should lead to four jets without a strong hierarchy in energy and without a distinctive mass scale aside from the jet mass. We simulate the signal with Pythia8.2.30 [29] and Delphes3.3.3 [30], as usually ignoring multi-parton interaction and pile-up. The fat jets are anti-k_T jets [32] with size $R = 0.8$, defined by FastJet3.2.2 [33] with

$$p_{T,j} = 475 \ldots 525 \text{ GeV} \,. \tag{10}$$

As before, the objects of the subjet analysis are particle flow objects [34] from the Delphes E-flow. The leptons from the charm decays are taken into account for the calorimeter. For

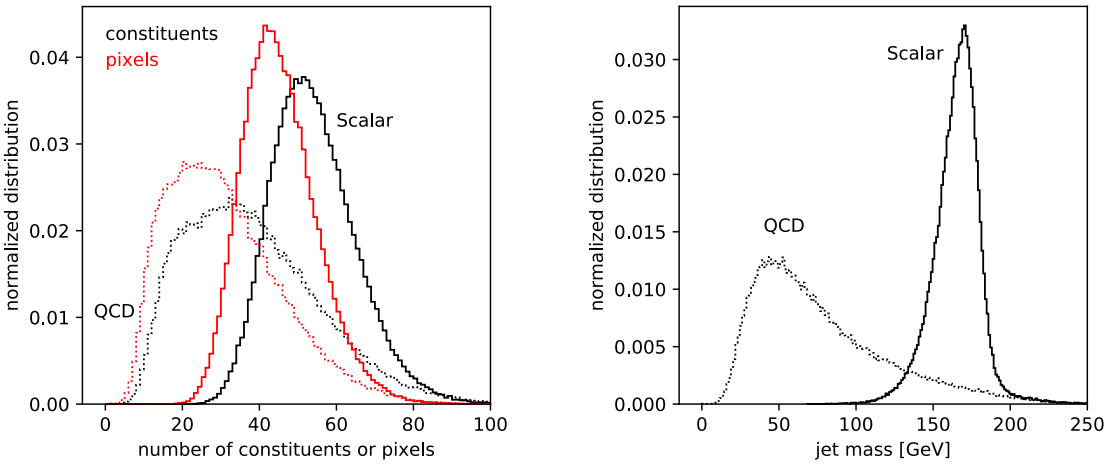

Figure 8: Left: numbers of constituents and of non-zero pixels for scalar decay jets and QCD. Right: truth-level jet mass distributions for the signal and the QCD background.

the pre-processing we center the jets in the k_T-weighted centroid before pixelization and use a range of -0.75 ...0.75 for the azimuthal angle and for the rapidity.

In Fig. 8 we show the main physics patterns of the scalar decay jets compared to the QCD background. In the left panel we see the number of constituents. Comparing to Fig. 1 we see that the general patterns are very similar, with the color-charged top leading to a slightly larger number of constituents. In the right panel we show the jet masses for the signal and the background. Both plots indicate that the heavy scalar signal is very similar to the top signal, but without the intermediate mass drop from the W-decay. This will force the de-correlated network to discriminate signal and background just based on the number of properties of the constituents from the scalar decay vs QCD radiation. In principle, we could increase the reach for this model by applying c-tagging, but for our toy model we explicitly do not want to use this additional information.

would help if the plots displayed the mean here

The setup of the autoencoder network with and without adversary is exactly the same as for the top case, including a bottleneck size of 32 units. In the left panel of Fig. 9 we include a ROC curve for the image-based autoencoder network without adversary, trained on QCD jets only. It corresponds to an AUC value of 0.90, comparable to the top case. As before, we can add an adversary to the autoencoder, to remove the information on the jet mass from the network and to generate control samples. This leads to a weaker performance of the network. For the same bottleneck of 32 units and a Lagrangian multiplier $\lambda = 10^{-3}$ we find the ROC curve given in Fig. 9 with an AUC value of 0.60. As mentioned before, this is significantly worse than for the top case, because the scalar is missing a second mass drop at intermediate masses.

To see the effect of the adversarial, we show the performance after training on pure QCD jets and evaluated on a sample including 3% signal jets in analogy to Fig. 6. Two sets of curves include all jets or the 5% least QCD-like jets in the right panel of Fig. 9. First, we indeed observe a small enhancement around $m_j = m_t$. While for our choice of the Lagrangian multiplier there remains a small background shaping, we also observe a clear signal enhancement for the least QCD-like events. However, the scalar example also shows the limitations

How significant is this really? It looks like you might need more MC stats Also, going to “least QCD like” makes it worse?

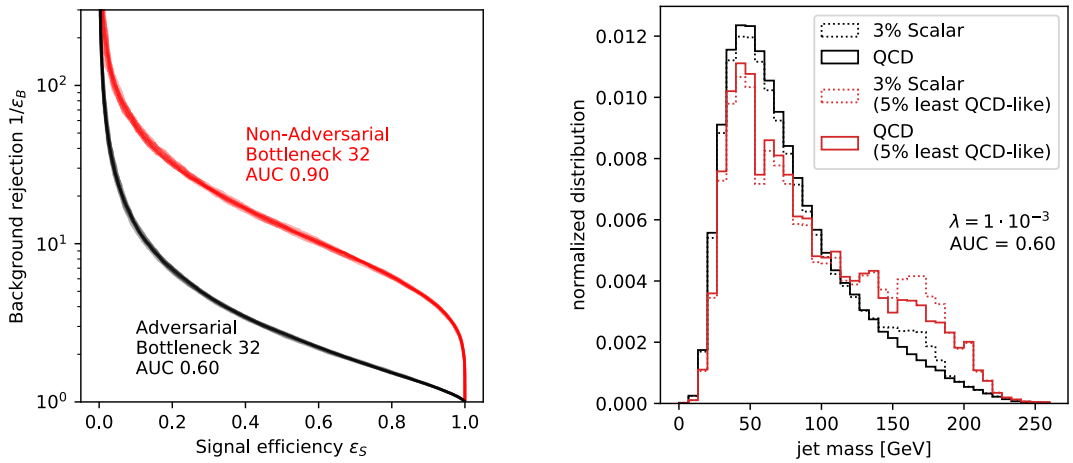

Figure 9: Left: ROC curves for the image-based autoencoders with and without adversary. Right: jet mass distributions from the adversarial autoencoder trained on pure QCD.

of a subjet analysis where we cannot apply a mass drop and have to rely on difference similar to quark-gluon discrimination.

3.2 Dark showers

order of references

We use modified, dark showers [40] as another benchmark scenario, independent of their new physics motivation through hidden valley models [39]. We assume that the model includes a heavy dark quark q_v which can be pair-produced at the LHC. It undergoes showering in the dark and SM sectors and eventually decays to its SM-partner and a light dark boson, b_v, which is uncharged under all SM-gauge groups. This dark boson hadronizes into scalar and pseudoscalar dark meson states, collectively labelled as π_v and assumed to have identical masses $m_{\pi_v} = 2m_{b_v}$. Depending on the model parameters the dark mesons can decay back to SM particles via a reverse of the production process, or leave the detector unobserved. The visible signature is therefore di-jets plus a variable amount of missing energy

$$pp \to q_v\bar{q}_v \to q\bar{q} + \not{E}_T \; . \tag{11}$$

However, the exotic production mechanism through a heavy color-charged dark quark leads to a sizeable amount of QCD radiation together with the dominant jets. It generates a jet mass spectrum with an upper edge at the dark quark mass. For our study we use a range of dark quark and dark boson masses. The dark gauge interaction we consider is $SU(3)_v$ with $\alpha_v = 0.1$, which is a Pythia default model.

The generation setup of for the dark showers is the same as for the heavy scalar in Sec. 3.1, only with a slightly higher p_T range,

$$p_{T,j} = 575 \; ... \; 625 \text{ GeV} \; . \tag{12}$$

The image preprocessing is identical to the scalar case with minimal pre-processing before pixelization.

For the dark shower model parameters we again ignore current experimental constraints and choose scenarios which best test and illustrate the behavior of our adversarial autoencoder.

around 0.6, but with better jet mass de-correlation. As expected, the mass peak for the 5% least QCD-like events is broader and less pronounced for the mass-degenerate model. As for the scalar case, we clearly see that the autoencoder strategy works, but also that most of the relevant information is included in the jet mass distribution. In return, de-correlating this key observable for background control leads to a significant drop in performance.

4 Outlook

Anomalies in jets at the LHC can be extracted with the help of an autoencoder, a neural network based on low-level data and trained on QCD or other background samples only. We have shown that such a network extracts boosted hadronic top decays based on jet images or based on 4-vectors with a simplified LoLa structure. Its reduced performance as compared to specialized taggers is balanced by reduced systematic uncertainties in the absence of a well-defined signal model. Moreover, one autoencoder network realizing un-supervised learning for a given phase space region can be used to search for many different signals at the same time.

To further reduce experimental systematics, we propose to train and use an autoencoder network in the same phase space region. This requires full control the background shaping. We extend our approach to an adversarial autoencoder based on jet images, de-correlating for example the jet mass from the training. This allows us to sort a jet sample by the loss function describing how QCD-like the jet is. We find (essentially) the same flat jet mass distribution for each slice in the loss function. For instance top decay jets are now collected in the least QCD-like slices and lead to a distinct peak in the jet mass.

Next, we have shown how to train the adversarial autoencoder on data with a signal contamination. In that case we typically make the autoencoder more restrictive and still find that the top jets are classified as the least QCD-like jets. We can still select them based on the network output and search for their distinctive peak in the jet mass distribution for non-QCD slices.

Finally, we have shown how the (adversarial) autoencoder can be used to not only extract top decay jets, but also decays of a heavy scalar to four quarks, or dark showers. Both of these models are significantly harder to extract than tops at the LHC. After de-correlating the jet mass, the different signals retain different amounts of information, allowing us to separate them from the QCD background. Given the universal structure of the autoencoder network this means that the experimental LHC collaborations could make their networks, trained on data, public and allow external groups to test if specific models would indeed be flagged as anomalies and are hence excluded.

While finishing this paper we heard of a similar, independent study, which is published in parallel to our work [41].

This confused me. You mean the QCD jets are flat but the top jets are peaked? Why isn't the QCD distribution in each slice falling like the original rather than flat? Obviously I've misunderstood something, sorry.

Acknowledgments

We would like David Shih and his group for the very friendly and constructive coordination. We are grateful to Michel Luchmann for help with the improved image pre-processing. Finally, we would like the BOOST conference series for the encouraging atmosphere, without papers like this might never be written.