# Searching for new physics with deep autoencoders

Marco Farina,[1,2] Yuichiro Nakai,[2] and David Shih[2]

[1]*C.N.Yang Institute for Theoretical Physics, Stony Brook, New York 11794, USA*
[2]*NHETC, Dept. of Physics and Astronomy Rutgers, The State University of New Jersey, Piscataway, New Jersey 08854, USA*

(Received 10 January 2019; accepted 27 March 2020; published 13 April 2020)

We introduce a potentially powerful new method of searching for new physics at the LHC, using autoencoders and unsupervised deep learning. The key idea of the autoencoder is that it learns to map "normal" events back to themselves, but fails to reconstruct "anomalous" events that it has never encountered before. The reconstruction error can then be used as an anomaly threshold. We demonstrate the effectiveness of this idea using QCD jets as background and boosted top jets and R-parity violating (RPV) gluino jets as signal. We show that a deep autoencoder can significantly improve signal over background when trained on backgrounds only, or even directly on data which contain a small admixture of signal. Finally, we examine the correlation of the autoencoders with jet mass and show how the jet mass distribution can be stable against cuts in reconstruction loss. This may be important for estimating QCD backgrounds from data. As a test case, we show how one could plausibly discover 400 GeV RPV gluinos using an autoencoder combined with a bump hunt in jet mass. This opens up the exciting possibility of training directly on actual data to discover new physics with no prior expectations or theory prejudice.

DOI: 10.1103/PhysRevD.101.075021

## I. INTRODUCTION

Deep learning is a hot topic in high energy physics. It has been applied to tagging boosted jets of various kinds [1–15], to quark/gluon discrimination [16–18], and to full event classification [19–21]. These are all examples of supervised learning where the training sets are labeled with truth information. More recently, people have been starting to explore forms of weakly supervised and unsupervised learning (see e.g., Refs. [22–33]). In some weak-supervision approaches, binary classification is attempted on a data sample with only imperfect labels, for instance using class proportions or mixed samples [22–24]. Or there have been recent attempts to train a machine-learning algorithm to learn the probability distribution of the background and then compare this to the data to discover new physics [29,31]. Another approach to weakly supervised anomaly detection is to extend bump hunts with machine learning [25,28]. Applications of deep learning in high energy physics do not stop at classification tasks; pileup removal [34], generative models [35], and many others (for a review and more references, we refer to Ref. [36]) have all been studied.

Although the LHC has performed hundreds, if not thousands, of searches for new physics since its inception, so far no definitive evidence for physics beyond the Standard Model has turned up. All the searches for new physics in the expected places (supersymmetry, composite Higgs, fourth generations, $Z'$s, etc.) have turned up empty. This strongly motivates methods to look for physics without as much top-down theory prejudice. We need more ways to discover the unexpected at the LHC, and here is where unsupervised machine learning comes into play.

In this paper, we study one promising avenue to perform open-ended searches for new physics at the LHC: anomaly detection with autoencoders and deep learning. An autoencoder [37] is a simple idea with various incarnations and many real world applications to anomaly detection, denoising [38], generative models [39], feature selection, and more. (For an introduction to autoencoders and their applications, see e.g., Refs. [40–42].) In its simplest form, it is a lossy algorithm that maps an input to a latent compressed representation and then back to itself. This is illustrated in the cartoon in Fig. 1. A measure for how well the autoencoder performs is the difference between input and output according to some distance metric—the "reconstruction error." For example, for images, it could be the pixelwise, summed mean-squared difference between input and output. Typically, one trains an autoencoder on a sample of background events with the objective of minimizing reconstruction error on the sample. In this way, it learns what background "looks like." Any anomaly

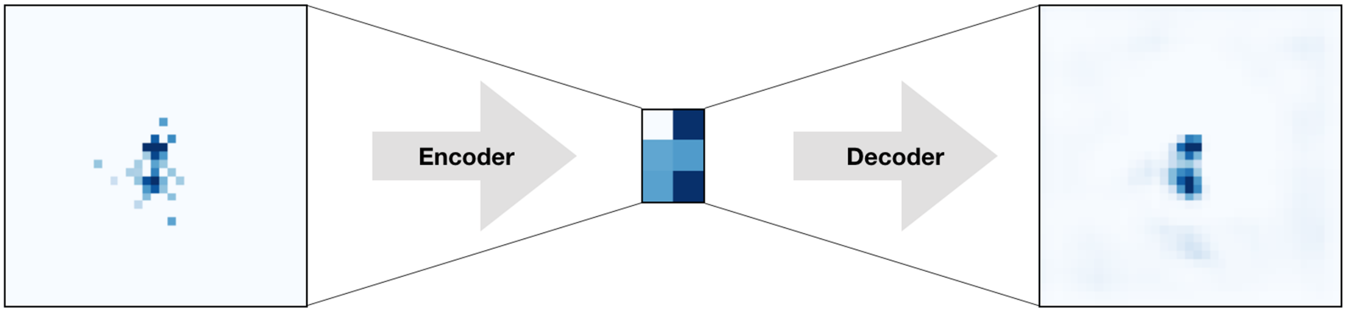

FIG. 1. The schematic diagram of an autoencoder. The input is mapped into a low(er)-dimensional representation, in this case six dimensions, and then decoded.

(the signal, e.g., new physics) is then expected to be poorly reconstructed by an autoencoder optimized on a sufficiently different background. Hence, we can use a cut on the reconstruction error as an anomaly threshold.

For concreteness, we will focus in this work on distinguishing "fat" QCD jets from other types of heavier, boosted resonances decaying to jets. Building on previous work on top tagging [12], we will concentrate on machine-learning algorithms that take jet images as inputs. For signal, we will consider all-hadronic top jets, as well as 400 GeV gluinos decaying to three jets via R-parity violating (RPV). Obviously, this is not meant to be an exhaustive study of all possible backgrounds and signals and methods but is just meant to be a proof of concept. The idea of autoencoders for anomaly detection is fully general and not limited to these signals. We will comment on other forms of inputs in Sec. V. Moreover, there are many other anomaly detection techniques that are not based on autoencoders and/or on reconstruction (loss) which are worth exploring in future work. For instance, see Ref. [32] for a different approach to anomaly detection that combines supervised feature learning with autoencoders for dimensionality reduction and clustering. Autoencoders have also been recently used in other high energy physics applications: in parton shower simulation [30] and for automated detection of detector aberrations in CMS [33].

We will explore various architectures for the autoencoder, from simple dense neural networks to convolutional neural networks (CNNs), as well as a shallow linear representation in the form of Principal Component Analysis (PCA). We will see that, while they are all effective at improving signal over background ($S/B$) by factors of $\sim$10 or more, they have important differences. The reconstruction errors of the dense and PCA autoencoders correlate more highly with jet mass, leading to greater $S/B$ improvement for the 400 GeV gluinos compared to the CNN autoencoder. While this may seem better at first glance, we discuss how one might want to use an autoencoder that is decorrelated with jet mass, in order to obtain data-driven sideband estimates of the QCD background and perform a bump hunt in jet mass. Indeed, we show how cutting on the reconstruction error of the CNN autoencoder results in stable jet mass distributions, and we show how this can be used to improve $S/B$ by a factor of $\sim$6 in a jet mass bump hunt for the 400 GeV gluino signal.

We will study the performance of the autoencoder in two modes: a version where it is trained on background-only events and a version where it is trained on a mixed sample containing both background and signal, meant to be representative of the actual data. An autoencoder trained on a sample of background-only events is an example of weakly supervised machine learning. One could still imagine applying this directly to data, provided one can prepare a control sample that consists only of representative backgrounds. Or one could train on Monte Carlo (MC) backgrounds and hope that the MC is an accurate representation of the background events in the data. As a first test of this assumption, we will train on PYTHIA and evaluate on both PYTHIA and HERWIG, and we will see that the results are similar.

By contrast, the autoencoder trained on mixed samples of background and signal is an example of fully unsupervised machine learning, and as such is a much more exciting potential application. We will show that, surprisingly, the autoencoder performance is remarkably stable against signal contamination; the performance is barely degraded even if signal is 10% of the training sample. Evidently, there is not much difference between the weakly supervised and fully unsupervised modes. Somehow, the autoencoder learns to preferentially reconstruct the background, and still poorly reconstructs the signal, even though it sees the signal as part of the training process. This raises the exciting possibility that the autoencoder could be trained directly on the data, and then could potentially discover any anomalous signal of new physics in the background (perhaps when combined with other variables, for instance a mass cut or bump hunt), provided it looks different enough from Standard Model (SM) objects. This would be an ideal method to discover the unexpected or to perform open-ended searches for new physics at the LHC.

Aside from open-ended anomaly detection, the autoencoder could be viewed as a general-purpose background cleaner. That is, we could train it on the background (or directly on the data) and then cut on reconstruction loss in order to remove "boring" QCD events, leaving behind a sample that is presumably more signal rich. We could then study these events in more detail, using other techniques and variables to isolate the signal.

We stress that using an autoencoder to search for new physics involves two different and important concepts in data analysis and machine learning: dimensionality reduction and anomaly detection. The novelty of the approach described in this paper is combining a dimensionality reduction method optimized on the training data (the encoder) together with a way of measuring the quality of the dimensionality reduction (the decoder and the reconstruction loss) and using this for anomaly detection. In other words, the autoencoder is not merely a method of dimensionality reduction; it also learns to decode the latent space back to the original space, which is at the heart of the anomaly detection method. Moreover, it accomplishes both in an unsupervised and data-driven way. In principle, one could imagine accomplishing the dimensionality reduction in other ways, e.g., more physics motivated, such as the basis of $N$-subjettiness variables described in [9]. However, they do not necessarily come equipped with an obvious inverse mapping; nor are they optimized on the data. While it would be interesting to explore using other dimensionality reduction techniques for anomaly detection, we believe this is the first attempt to do so in the literature.

The outline of the paper is as follows. In Sec. II, we define autoencoders quantitatively and present the architectures employed in the rest of the paper. We also describe the details of event generation used to obtain the data sets. Section III is devoted to the main results of the weakly supervised mode (with pure background training set). We compare the performance of the different architectures, discuss the methods by which we choose the size of the latent space, and perform a MC comparison in the form of PYTHIA VS HERWIG. In Sec. IV, we turn our attention to the fully unsupervised mode. We study the consequences of having a small fraction of signal in the training set, and then we discuss correlation between jet mass and reconstruction loss of the trained autoencoders. We show how by using the CNN autoencoder, a bump hunt in jet mass could potentially reveal the presence of 400 GeV RPV gluinos in the actual data. Finally, we conclude in Sec. V with a summary and list of future directions.

## II. METHODS

Let us start with a more detailed introduction to autoencoders. Given an input $x \in \mathbb{R}^n$, we want to learn a mapping into $\hat{x} \in \mathbb{R}^n$ while passing through a latent representation $y \in \mathbb{R}^k$. This mapping is implemented by two functions: the encoder $f: \mathbb{R}^n \to \mathbb{R}^k$ and the decoder $g: \mathbb{R}^k \to \mathbb{R}^n$. The functional forms of $f$ and $g$ are determined by the autoencoder architecture; they are parametrized by sets of learnable weights, $\theta_f$ and $\theta_g$, respectively. The aim of the autoencoder (and the aim of the machine-learning training process) is to ensure that $x$ and $\hat{x} = g(f(x;\theta_f);\theta_g)$ are as close as possible under a given metric. Useful results are obtained when the dimension of the latent space is smaller than the input one, $k \ll n$, so that the trivial mapping cannot be learned. Thus, the autoencoder learns a compressed representation of the input, optimized on its features.

To evaluate the distance between $x$ and $\hat{x}$, we will use the $L^2$ norm, also known as the mean-squared reconstruction error:

$$L(x,\hat{x}) = \frac{1}{n}\sum_{i=1}^{n} |x_i - \hat{x}_i|^2. \tag{2.1}$$

By training the autoencoder to minimize $L$ on a sample of background events, we learn to encode and decode the typical events that arise from the background distribution. Then, when the autoencoder is evaluated on signals that do not come from the background distribution, the hope is that it will result in a larger $L$ than usual. Thus, the tail of the $L$ distribution is more likely to be signal than background, and by cutting on $L$, we can cut out background and better detect signals. This one of the possible ways to use an autoencoder as anomaly detector.

### A. Sample generation

The jet image samples used in this work follow the exact same specifications as the "CMS jets" used in Ref. [12]. We describe this briefly here, but we refer the reader to Ref. [12] for more detailed information.

The jets are generated using PYTHIA8.219 [43] for hadronization and DELPHES3.4.1 [44] for detector simulation. All jets are clustered with FASTJET3.0.1 [45]. We use anti-$k_T$ jets with $R = 1$, and we require $p_T \in [800, 900]$ GeV and $|\eta| < 1$.

The background (used for training the autoencoders) consists of light QCD jets, while for examples of signal, we will employ top quark jets and gluino jets with mass $m_{\tilde{g}} = 400$ GeV. The tops are assumed to decay hadronically, while the gluinos decay to three light-quark jets via RPV supersymmetry. All the samples are generated by simulating pair production of the heavy resonance starting from $pp$ collisions at 13 TeV (LHC Run II conditions).

In order to ensure that the decay products of the heavy resonance are predominantly contained within the fat jet, we apply a merge requirement of $\Delta R < 0.6$ at the truth level on the partonic daughters of the decayed heavy resonance. We also require a geometric match requirement of $\Delta R < 0.6$ between the fat jet and the original heavy resonance.

In all of our studies, we use sample sizes of 100,000 for training and testing. We have checked with smaller sample sizes that the performance of the autoencoders seems to saturate at 100,000, but we have not performed a detailed study.

After generating the fat jets, we apply several preprocessing steps described in Ref. [12] (center, rotate, flip, normalize), and then we pixelize the jets into 37 × 37 images whose pixel intensities correspond to total $p_T$. We stick to grayscale images in this work for simplicity.

### B. Autoencoder architectures

In this work, we compare two deep-learning autoencoder architectures, as well as a simpler autoencoder based on PCA that could be considered as a baseline. All of our autoencoders take the jet images as inputs. In the Appendix, we will provide full descriptions in the form of KERAS code. In this sub-section, we will describe them briefly and qualitatively:

(i) For preliminary exploration, we will use PCA. The principal components correspond to the eigenvectors of the correlation matrix, ordered in decreasing eigenvalues. The encoder projects the inputs onto the first $k$ principal components, and the decoder embeds the first $k$ principal components back into the original space. It can be shown that this minimizes the mean-squared error in the space of linear projections. Thus, in this sense, PCA is comparable to a linear model (e.g., one layer with linear activations and $k$ the dimension of the latent space) with the convenient property of being deterministic.

(ii) The simplest architecture we consider is just a series of dense (fully connected) layers. One starts by flattening the $N \times N$ image into a single column vector of length $N^2$. This is then fed to the dense layers of successively smaller size until one arrives at the latent layer. Then, this process is reversed until one arrives back at a column vector of the initial size.

(iii) For a more sophisticated autoencoder, we consider a CNN. Here, the dimensionality reduction is accomplished via the usual max-pooling layers. After a series of convolutional and max-pooling layers, the output is fed to a series of dense layers, resulting finally in the latent representation. The entire process is reversed [with two-dimensional (2D) upsampling layers in place of the max-pooling layers] to arrive back at an image with the same dimensions. (For the arithmetic of the max pooling and upsampling to work out, we zero-pad the inputs to the CNN autoencoder so that they are 40 × 40 pixels.)

All the architectures have been implemented using KERAS2.1.5 with TENSORFLOW1.7.0 backend on Nvidia GPUs (Pascal 100 and GeForce GTX 1080). For training, we used the default Adam algorithm with minibatch size of 1024[1] and a mild early stopping criterion: threshold = 0 and patience = 3 (= 5) for the CNN (dense) autoencoder. As this is a proof-of-concept paper, we have not optimized heavily the training algorithm (e.g., we have not studied the effect of learning rate annealing or momentum).

## III. TRAINING ON BACKGROUNDS: WEAKLY SUPERVISED MODE

We now present our results for each autoencoder described in the previous section. In this section, we study the weakly supervised case with pure background events for training, leaving the unsupervised case with samples contaminated by a small fraction of anomalous events to the next section.

### A. Autoencoder performance

Shown in Fig. 2 are histograms of the reconstruction errors computed with a CNN autoencoder, and $k = 6$ latent dimensions, for the background sample of QCD jets and the two different signals we consider in this paper (tops and gluinos). We see that the autoencoder works as advertised: it learns to reconstruct the QCD background that it has been trained on (to be precise, we train on 100,000 QCD jets, and then we evaluate the autoencoder on a separate sample of QCD jets), and it fails to reconstruct the signals that it has never seen before. For comparison, the typical per-pixel variance is $\mathcal{O}(10^{-2})$ (keep in mind that the jet images have been normalized to unit total pixel intensity), so we expect a baseline of $\mathcal{O}(10^{-4})$ for the reconstruction error.

This is further illustrated in Fig. 3, which shows the average QCD, top, and gluino jet image before and after autoencoder reconstruction. We see by eye that the QCD images are reconstructed well on average, while the others contain more errors.

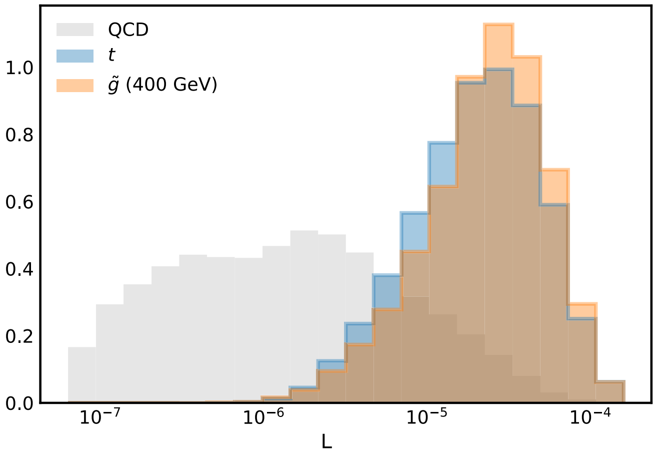

FIG. 2. Distribution of reconstruction error computed with a CNN autoencoder with $k = 6$ latent dimensions on test samples of QCD background (gray) and two signals: tops (blue) and 400 GeV gluinos (orange). For comparison, the typical per-pixel variance is $\mathcal{O}(10^{-2})$ (keep in mind that the jet images have been normalized to unit total pixel intensity), so we expect a baseline of $\mathcal{O}(10^{-4})$ for the reconstruction error.

[1] We found that a smaller minibatch size resulted in worse performance—the autoencoder converged too quickly and then overtrained.

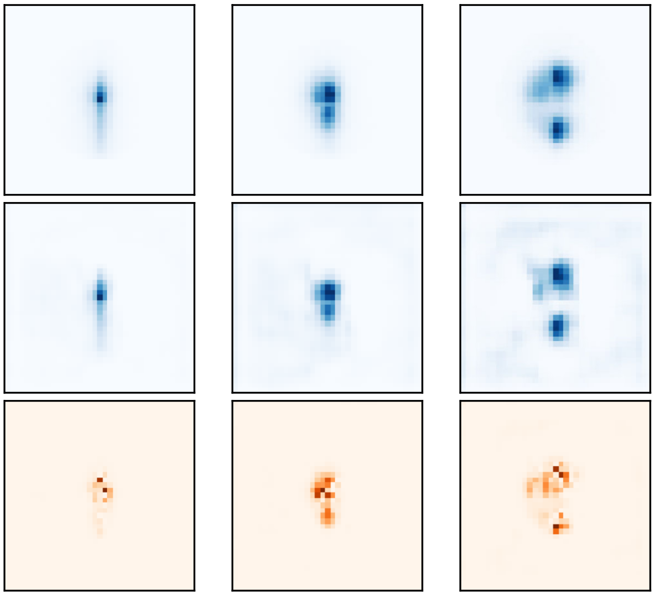

FIG. 3. Each panel represents the average of 100,000 jet images. Pixel intensity corresponds to the total $p_T$ in each pixel. Upper row: original sample. Middle row: after reconstruction. Lower row: pixelwise squared error. Left column: QCD jets. Middle column: top jets. Right column: $\tilde{g}$ jets. The reconstructions have been computed with a CNN autoencoder and $k = 6$ latent dimensions.

By sliding the reconstruction loss threshold $L > L_S$ around, we can turn the histograms in Fig. 2 into receiver operating characteristic (ROC) curves. The ROC curves for the different autoencoder architectures are shown in Fig. 4 for the top and gluino signals. For comparison, we have also included the ROC curve obtained by cutting on jet mass as an anomaly threshold. For dense and CNN, we show the performance of one run, given the small variance between each run of training. While the three architectures have comparable performances, it is clear there are some important differences. For tops, the CNN outperforms the others, while for gluinos, the situation is largely reversed. Surprisingly, for gluinos, the CNN is even outperformed by the humble PCA autoencoder at all but the lowest signal efficiencies. We will explore this in more detail in Sec. IV B, but a clue as to what is going on is shown in the comparison of the PCA ROC curve with the jet mass ROC curve. For gluinos, they track each other extremely closely, suggesting that the PCA reconstruction error is highly correlated with jet mass. We will confirm this in Sec. IV B. Evidently, the PCA autoencoder (and to a lesser extent the dense autoencoder) has learned to reconstruct the more numerous low mass QCD jets at the expense of the rarer high mass QCD jets. In this sense, the PCA autoencoder has not learned to reconstruct the mass well. Meanwhile, the CNN reconstruction loss is less correlated with jet mass at higher jet masses (again, see Sec. IV B). This is evidence that the CNN has learned to better reconstruct jet mass. This is to be expected, given the higher expressive power of CNNs.

In Table I, we show the signal efficiency at 90% and 99% background rejection (which we refer to as $E_{10}$ and $E_{100}$, respectively). The values reported in each case are the average over five independent training runs to ameliorate the intrinsic variance (apart from PCA, which is deterministic). We see that rejecting 99% of background will keep more than 10% of the signals for both of the deep-learning-based autoencoders.

### B. Choosing the latent dimension

Here, we will explore the dependence of the autoencoder on the dimension of the latent space. This is one of the most

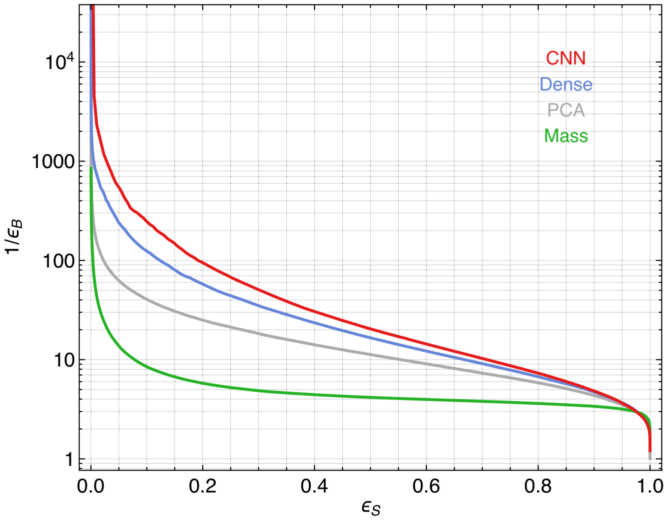

FIG. 4. ROC curves of tagging efficiency $\epsilon_S$ vs background rejection $1/\epsilon_B$ computed with a CNN autoencoder and $k = 6$ latent dimensions on test samples consisting of top jets (left) and gluino jets (right). These ROC curves are for a single training of the autoencoder; we have checked that the variance from training to training is sufficiently small so as to make a negligible difference to the plot.

TABLE I. $E_{10}$ and $E_{100}$ values for various signals. Results for dense and CNN, with $k = 6$ latent dimensions, are obtained as the average of five runs of training on the 100,000 sample (the variances are at the ~0.01 level).

| | $t$ | $\tilde{g}$ |
|---|---|---|
| PCA | 0.51/0.04 | 0.98/0.36 |
| Dense | 0.66/0.13 | 0.90/0.39 |
| CNN | 0.70/0.19 | 0.77/0.23 |

important choices to make in the design of an autoencoder for anomaly detection. If the dimensionality is too low, the autoencoder is not able to capture all the salient features of the training set. On the other hand, as the encoding space gets larger, we get closer to the trivial representation. Hence, we would like to find an optimal compromise.

In choosing the latent dimension of the autoencoder, it is important to keep in mind the unsupervised nature of our endeavor. So, optimizing the latent dimension using various signals is not the approach we want to take.

One unsupervised method for finding an optimal working point is to use PCA as the initial step. Shown in Fig. 5 (left) is the amount of variance in the data explained by each eigenvector of PCA, in descending order. (This kind of plot is conventionally referred to as a "scree plot" by PCA practitioners who also happen to be mountaineers.) An obvious and common prescription is to choose the number of principal components close to the "elbow" of the scree plot; other choices might be motivated upon more detailed inspection of the cumulative accounted variance (e.g., one might choose the number of encoding dimensions corresponding to 95% or 99% of the total variance). We could then use the same value for the dimensionality of the encoding space in our deep networks.

We can also search for a similar behavior in the loss function. This is shown in Fig. 5 (right) for the different autoencoders. We see the loss plateaus around the same place for the various autoencoders, and that corresponds roughly to the elbow of the PCA scree plot. The loss function first sharply decreases as more important and meaningful features are learned by the encoded representation. It reaches a plateau supposedly when only marginal information is added to the encoding space.

Following the above logic, we choose $k = 6$ encoding dimensions for all of the autoencoders presented in the paper.

Finally, let us examine the wisdom of our choice by looking at the top signal for example. Shown in Fig. 6 is $E_{10}$ and $E_{100}$ for the top signal (averaged over five training runs) as a function of the latent dimension. This shows the same behavior as we saw above—the performance of the autoencoders plateau around $k = 6$. This is encouraging evidence for our unsupervised method of choosing the latent dimension based on PCA and reconstruction loss.

### C. Robustness with other Monte Carlo

Before turning to unsupervised approaches in the next section, let us consider here the main weakness of the weakly supervised approach: the reliance on accurate background-only samples for training.

One data-driven approach would be to define a control sample of fat jets, e.g., by inverting a lepton selection. This of course assumes the signal is never produced in association with leptons.

Alternatively, one would train on background Monte Carlo, and then apply the autoencoder to data. This would work only insofar as the Monte Carlo accurately represents the background in the data, or that any artifacts special to the Monte Carlo are not learned by the autoencoder. In particular, different hadronization schemes

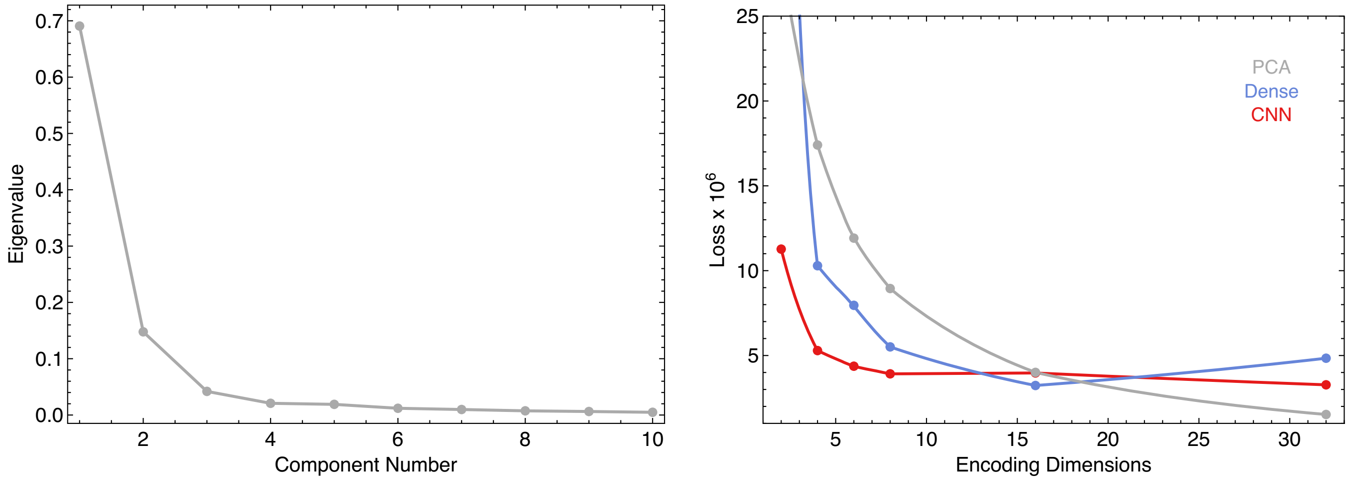

FIG. 5. Left: Scree plot for PCA. Contribution to the variance of each principal component in descending order. Right: average loss as a function of encoding space dimensions. Each dot corresponds to the average of five independent training runs on the 100,000 training sample (apart from PCA, which is deterministic and has no variance).

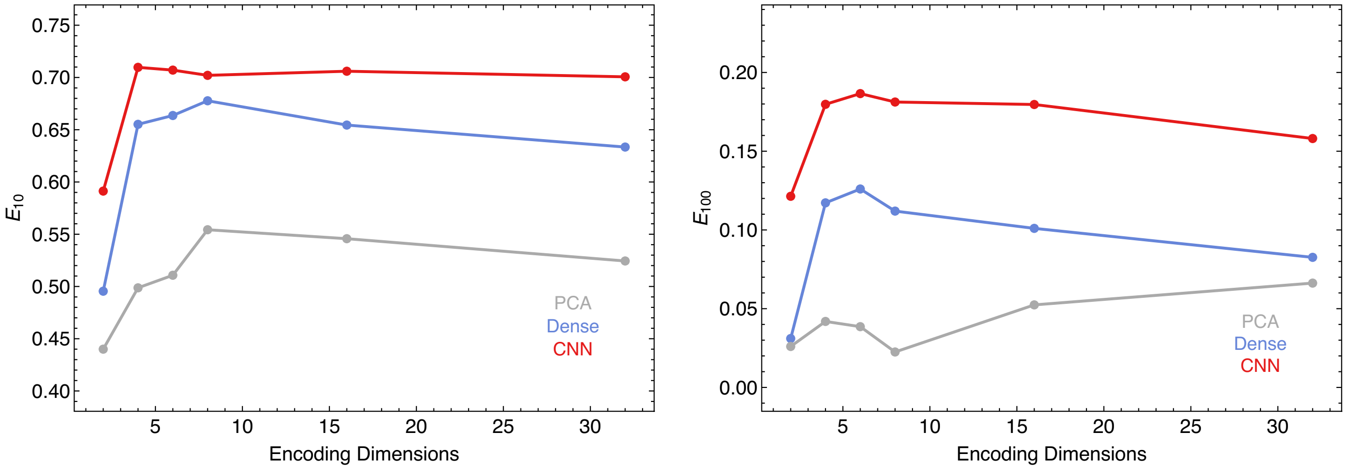

FIG. 6. Dependence of performance of autoencoders in the weakly supervised learning on number of dimensions of latent space. The values of $E_{10}$ and $E_{100}$ for top jet signals are shown, respectively, in the left and right panels. Each dot corresponds to the average of five independent training runs on the 100,000 training samples (apart from PCA, which is deterministic and has no variance).

could have an impact on the final shape of the jets we study and deteriorate the results of an autoencoder.

In this subsection, we will explore the dependence of the autoencoder on the choice of MC generator by evaluating our CNN autoencoder (trained on PYTHIA) on fat jets produced with HERWIG. Figure 7 shows the resulting distributions of the reconstruction error. The differences are small, and crucially the separation between background and anomaly is preserved. This can be seen as another proof that the autoencoder has mostly learned fundamental jet features which should depend only weakly on the hadronization scheme details.

We can quantify the degradation in performance by fixing a common threshold. For convenience, we choose it such that on PYTHIA we have the usual 90% and 99% background rejection. We select one training instance of the CNN autoencoder at random, which corresponds to $E_{10} = 0.71$ and $E_{100} = 0.19$. Applying the same threshold and the same algorithm to the HERWIG set, we obtain precisions of $\epsilon_s = 0.74$ and $\epsilon_s = 0.21$, respectively, with corresponding background rejection of 87% and 98%.

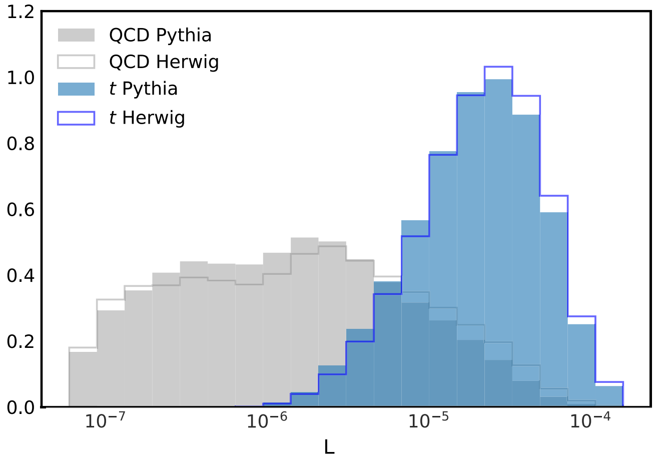

FIG. 7. Comparison of reconstruction error distributions between PYTHIA and HERWIG generated test samples, full colored histograms, and outlines, respectively. Gray is QCD and blue tops. The results are obtained after training a CNN on the PYTHIA train dataset.

## IV. TRAINING DIRECTLY ON DATA: UNSUPERVISED MODE

### A. Contamination study

In the previous section, we have explored how autoencoders can be trained on samples of background-only jets, and then be used to discover signals such as top quarks and RPV gluinos. This is a prime example of "one-class classification" and weakly supervised learning. It could potentially have direct applications to LHC searches for new physics, provided the background sample can be validated somehow.

In this section, we will turn to a potentially much more exciting application of autoencoders in the form of unsupervised learning. Rather than train on a sample of background-only jets, we will train on a sample of backgrounds "contaminated" by a small fraction of signal events. We will see how, somewhat surprisingly, the autoencoder still succeeds in detecting anomalies in the test set even though they are present in the training set. Evidently, as long as the autoencoder does not see "too many" anomalies in the course of its training, its performance will be largely preserved.

Figure 8 shows how the amount of contamination with anomalous events in the training set affects the performance of autoencoders. Here, we use top jet samples for anomalous events. The horizontal axis denotes the fraction $f_{\mathrm{cont}}$ of top jets in the entire training set. In the left and right panels, the values of $E_{10}$ and $E_{100}$ for top jet signals are shown,

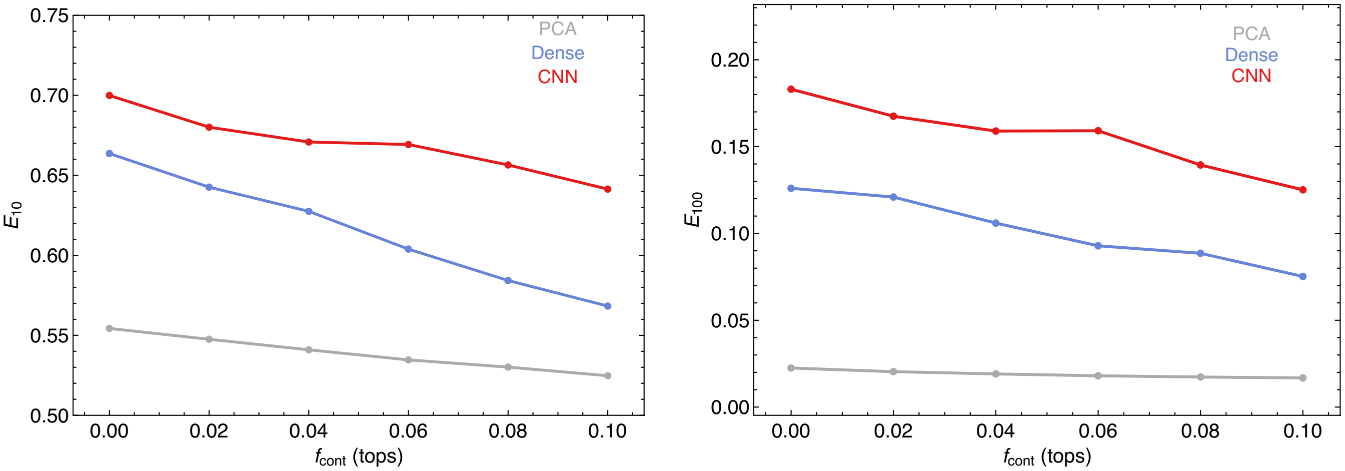

FIG. 8. The performance of autoencoders in the unsupervised learning case where the training set is contaminated with anomalous events. We take top jet samples for anomalous events. The horizontal axis denotes the ratio $f_{\rm cont}$ of top jet samples in the whole training set with 100,000 samples. In the left and right panels, the values of $E_{10}$ and $E_{100}$ for top jet signals are shown, respectively. The gray, blue, and red curves denote the cases of the PCA, dense, and convolutional autoencoders (each dot representing the average of five runs).

respectively. For dense and CNN autoencoders, each point represents the average of five runs. In every architecture, as the contamination ratio increases up to 0.1, the values of $E_{10}$ and $E_{100}$ tend to gradually decrease, but the reduction is not dramatic. This indicates that the contamination does not give a significant impact on the performance of our autoencoders.

Shown in Fig. 9 is a similar comparison for contamination with gluinos. We see that at fixed background rejection, the signal efficiency decreases by 10%–20% as the contamination fraction of the training sample is increased from 0% to 10%.

Just to emphasize how powerful this method potentially is, we see that with the CNN autoencoder, even with 10% signal present in the training sample, the autoencoder arrives at $E_{100} \sim 0.1$, so after this cut on reconstruction loss, we would end up with $S/B \sim \mathcal{O}(1)$.

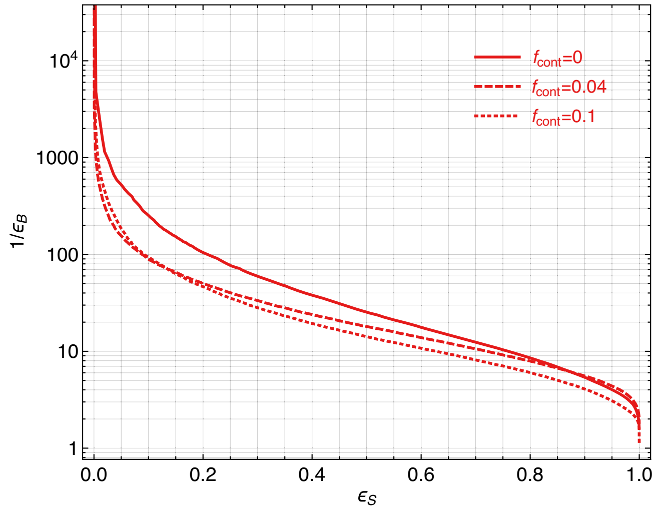

FIG. 9. ROC curves for CNN autoencoders trained on samples of QCD events contaminated with a fraction $f_{\rm cont}$ of gluino events.

Of course, without some way of estimating the background, this unsupervised method of searching for new physics would still probably have limited utility. With just a pure counting experiment (counting the number of events above some reconstruction error threshold), we would have no way of knowing whether we have found new physics, unless we knew beforehand what to expect from the SM background. In the next subsection, we will explore the possibility of combining the autoencoder with a variable like jet mass, in order to perform a bump hunt, with data-driven background estimates coming from sidebands.

### B. Correlation with jet mass

In this subsection, we will explore the correlation of the different autoencoders with jet mass.[2] We are motivated by how the autoencoder would be applied in the real world to look for new physics. We are looking for subtle signals in an open-ended way buried in the QCD background. Given that there is no reliable way to estimate the QCD background other than data-driven methods, and given that we are not expecting to achieve extremely high $S/B$ significances, a pure counting experiment seems implausible. Instead, we will still need another variable to sideband in order to estimate the QCD background from the data. Since a large class of new physics starts from the decay of a heavy new resonance, jet mass is an obvious candidate to sideband in.

[2] Since we are studying a narrow range of $p_T$'s in this paper and we have normalized away the total $p_T$ of each jet image, it may be more correct to say that we are studying the correlation with $m/p_T$. Indeed, we have checked for several values of $m$ and $p_T$ that the effect is similar.

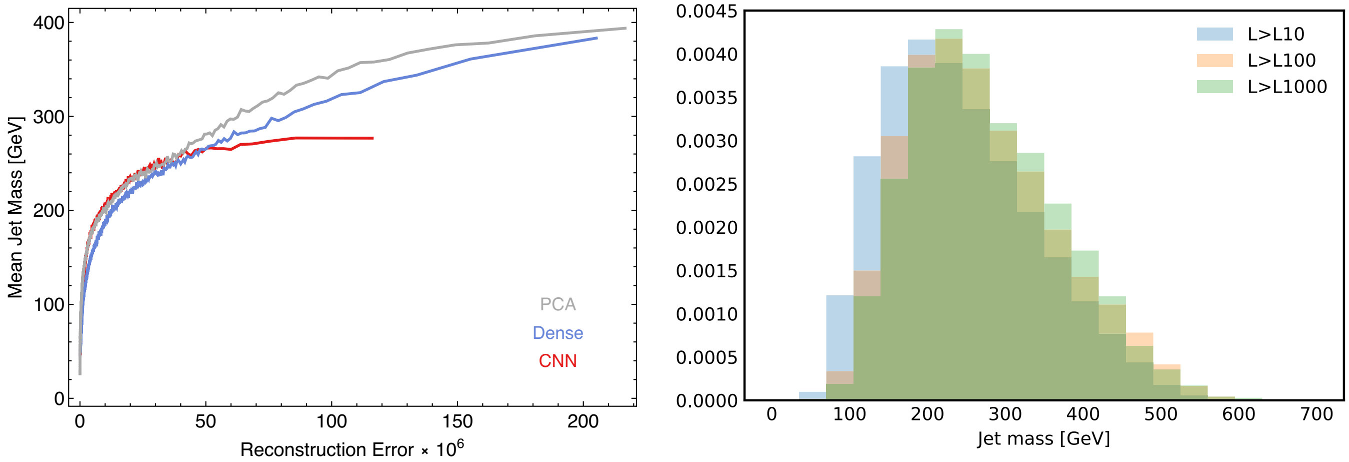

FIG. 10. The left figure shows the average mass in bins of increasing reconstruction error, for the different autoencoder architectures. We see that the PCA and dense autoencoder losses are highly correlated with jet mass all the way up to 400 GeV, while the CNN becomes uncorrelated for masses above ~300 GeV. The right figure illustrates this with jet mass histograms for the QCD background. We see that they are stable against increasingly hard cuts on the reconstruction error.

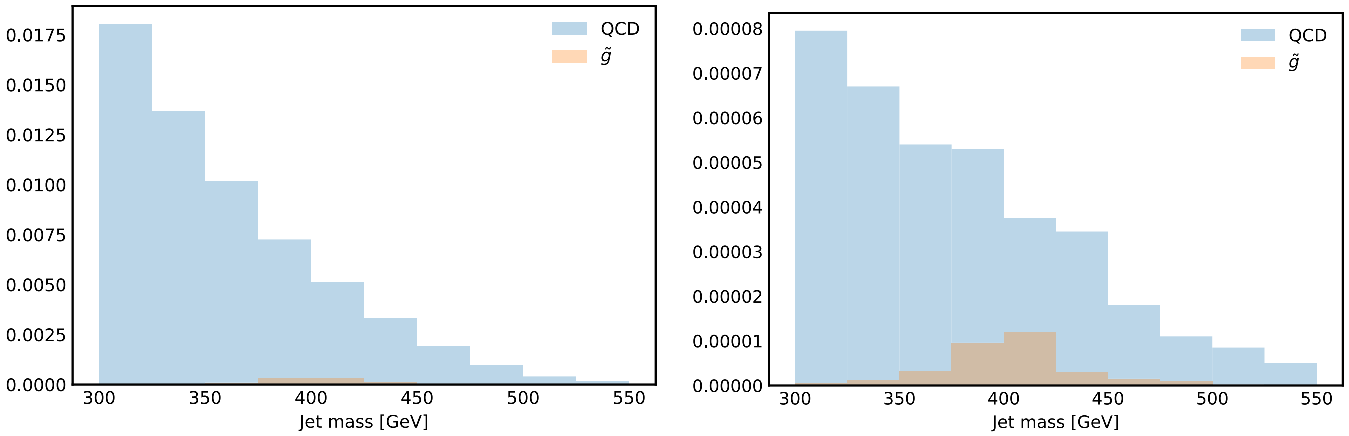

FIG. 11. Jet mass histograms for QCD background and 400 GeV RPV gluinos, normalized to their LO cross sections, before (left) and after (right) a cut on CNN autoencoder loss that rejects a factor of 1000 of the QCD background.

From this point of a view, the ideal autoencoder would be one whose reconstruction error is minimally correlated with jet mass. We could then cut hard on the reconstruction error to "clean" out the QCD background, and then look for a bump in the jet mass distribution, confident that the autoencoder cut did not sculpt an artificial peak into the jet mass distribution of the QCD background.

Shown in Fig. 10 (left) is the mean jet mass computed in bins of increasing autoencoder loss, for the QCD background. We see that PCA (gray) and dense (blue) reconstruction errors are correlated with jet mass all the way up to 400 GeV. So, cutting on the PCA loss is roughly equivalent to cutting on the jet mass. However, for CNNs, the correlation stops for jet masses above ~250–300 GeV. Equivalently, the jet mass distribution should be stable against cutting on the CNN loss for cuts above $\sim 10^{-6}$.

This is borne out in Fig. 10 (right). Here, we see the jet mass distribution after cuts on CNN loss that reduce the QCD background by a factor of 10 (blue), 100 (orange), and 1000 (green). The jet mass distribution is remarkably stable as we cut harder on CNN loss. This makes it the superior autoencoder for doing a bump hunt in jet mass for jet masses above ~300 GeV.[3]

To illustrate the possibilities of searching for new physics in this way, by first "cleaning" the QCD background using the CNN autoencoder and then doing a bump hunt in jet mass, we include Fig. 11. These are the jet mass histograms for QCD background and 400 GeV gluinos, now

[3] We note that a better approach would probably be to explicitly decorrelate the autoencoder output with jet mass, e.g., using an adversarial network. This would be interesting to explore further (in fact, see Ref. [46]), but it is beyond the scope of this work.

normalized to the leading order (LO) gluino and QCD cross sections, before (left) and after (right) a cut on CNN autoencoder loss that removes a factor of 1000 of the QCD background. Importantly, we have trained the autoencoder on a mixed sample containing the expected fraction of gluino jets, corresponding to an overall contamination fraction of $10^{-3}$. This would be representative of the actual data, if it really contained these gluinos.

We see that the $S/B$ achievable here in a mass window around the gluino mass is ≈25%. As can be seen clearly from the histograms, this is an impressive improvement on the $S/B$ before the cut (i.e., just from the raw jet mass histogram), which is only ≈4%. This improvement in $S/B$ could be important in situations where $S/B$ is small and we are limited by systematic and not statistical errors.

We can similarly quantify the gain in statistical significance. According to the ROC curve in Fig. 4 right (again, the ROC curve for unsupervised learning with this small amount of contamination will be very similar), the significance improvement $\epsilon_S/\sqrt{\epsilon_B}$ is approximately a factor of 1.25 at this working point. At working points with higher efficiency, it is as much as a factor of 2–3. One could plausibly discover new physics this way.

## V. CONCLUSIONS

In this paper, we have shown how autoencoders—machine-learning algorithms that learn how to compress and decompress a sample of inputs—are potentially powerful new tools for performing open-ended searches for new physics at the LHC. While autoencoders have many real-world applications to anomaly detection, they have up till now not been widely adopted in high energy physics.

We explored autoencoders in both weakly supervised and unsupervised forms. In the former mode, we trained autoencoders based on dense and convolutional neural networks on a sample of high $p_T, R = 1$ QCD jet images and showed how they could learn to accurately reconstruct these jet images. Then, the hope of using autoencoders for open-ended anomaly detection is that it would fail to reconstruct signals it had not been trained on, and then one could use the reconstruction error as an anomaly threshold. In this paper, we demonstrated that the deep autoencoders work as advertised, by applying them to signals consisting of all-hadronic top jets and RPV gluinos. We saw that by thresholding on reconstruction error, the autoencoder could improve $S/B$'s on these signals by sizable amounts.

We also showed how the autoencoder could operate in an unsupervised mode, and discover signals despite having been trained on data that actually contained those signals. In fact, we saw that varying the signal fraction even up to 10%, the autoencoder performance was remarkably stable. This implies that one could simply train the autoencoder directly on the data, and then look for a feature corresponding to new physics. As a proof of concept, we showed how this could be done with a jet mass bump hunt. We showed that the CNN autoencoder is reasonably decorrelated with jet mass, meaning that we could use the autoencoder to reduce the QCD background and then search for a bump in the jet mass distribution. We saw that it could achieve $S/B \sim 25\%$ for a 400 GeV RPV gluino signal, an improvement of over a factor of 6 from the bump hunt without autoencoder.

We believe this is a very exciting new direction in the search for new physics at the LHC, very unlike conventional approaches. There are many future directions that we envision. Some of these include:

(i) Testing out the autoencoder on other signals and backgrounds. For concreteness, we focused fat jets in a narrow range of $p_T$'s, treating QCD as background, and heavy resonances with three subjets as signal. But obviously, the idea is general and can be applied to any training and test samples in principle. One could envision applying this to other numbers of subjets, dark showers, nonresonant particles, etc.

(ii) Going further, it would be fascinating to train an autoencoder to flag entire events as anomalous, instead of just individual fat jets.

(iii) We focused on just a few autoencoder architectures in this paper, for the proof of principle, but there are many others on the market. For instance, recurrent neural networks originally designed for sequences and natural language processing. These have proven to be useful for boosted-object tagging [7,8,11,17], so we expect they will also be useful here. There are also even more complex types of anomaly detection in the computer-science literature based on the idea of generative adversarial networks (GANs) [47–49] that may also prove useful in this context.

(iv) It would be interesting to dive deeper into the latent representation that is learned by the autoencoder. Do signals and backgrounds cluster in this latent space? Do the latent dimensions correlate strongly with known variables such as jet mass and N-subjettiness?

(v) We saw here how the CNN autoencoder was reasonably decorrelated with mass. It would be interesting to explore ways to more explicitly decorrelate in mass. The "variable planing" ideas of Refs. [3,50] may be useful in this context. Or one could envision training an ensemble of autoencoders on jet samples corresponding to different bins in jet mass. A small enough bin width would probably ensure practical absence of correlation between mass and reconstruction loss. This is well beyond the scope of our study; we reserve this for future work.

Autoencoders are a form of weakly supervised or unsupervised machine learning which could be ideally suited to the current situation at the LHC, where many top-down-motivated searches have not turned up any evidence for new physics, and many people are wondering what we should be looking for. With an autoencoder approach, one does not need to know what one is looking for. It is a powerful new method to search for any signal of new physics in the data, without prejudice.

## ACKNOWLEDGMENTS

We thank Jack Collins, Kevin Nash, Marc Osherson, and Duccio Pappadopulo for helpful discussions. D. S. is grateful to the Aspen Center for Physics, which is supported by National Science Foundation, Grant No. PHY-1607611, where this work was performed in part. This work was supported by the DOE Grant No. DE-SC0010008.

*Note added*—Recently, we learned of the work of Ref. [46], which also studied the applications of autoencoders to anomaly detection and searching for new physics at the LHC.

## APPENDIX: KERAS CODE FOR AUTOENCODER ARCHITECTURES

### 1. Dense

```
input_img = Input (shape = (37 * 37, ))
layer = Dense (32, activation = ’relu’) (input_img)
encoded = Dense (6, activation = ’relu’) (layer)

layer = Dense (32, activation = ’relu’) (encoded)
layer = Dense (37 * 37, activation = ’relu’) (layer)
decoded = Activation (’softmax’) (layer)

autoencoder = Model (input_img, decoded)
autoencoder.compile(loss = keras.losses.mean_squared_error,
            optimizer = keras.optimizers.Adam())
```

### 2. CNN

```
input_img = Input (shape = (40, 40, 1))

layer = input_img
layer = Conv2D (128, kernel_size = (3, 3),
               activation = ’relu’, padding = ’same’) (layer)
layer = MaxPooling2D (pool_size = (2, 2), padding = ’same’) (layer)
layer = Conv2D (128, kernel_size = (3, 3),
               activation = ’relu’, padding = ’same’) (layer)
layer = MaxPooling2D (pool_size = (2, 2), padding = ’same’) (layer)
layer = Conv2D (128, kernel_size = (3, 3),
               activation = ’relu’, padding = ’same’) (layer)
layer = Flatten () (layer)
layer = Dense (32, activation = ’relu’) (layer)
layer = Dense (6) (layer)
encoded = layer

layer = Dense (32, activation = ’relu’) (encoded)
layer = Dense (12800, activation = ’relu’) (layer)
layer = Reshape ((10, 10, 128)) (layer)
layer = Conv2D (128, kernel_size = (3, 3),
               activation = ’relu’, padding = ’same’) (layer)
layer = UpSampling2D ((2, 2)) (layer)
layer = Conv2D (128, kernel_size = (3, 3),
               activation = ’relu’, padding = ’same’) (layer)
layer = UpSampling2D ((2, 2)) (layer)
layer = Conv2D (1, kernel_size = (3, 3), padding = ’same’) (layer)
layer = Reshape ((1, 1600)) (layer)
layer = Activation (’softmax’) (layer)
decoded = Reshape ((40, 40, 1)) (layer)
```

*(Table continued)*

*(Continued)*

```

    autoencoder = Model(input_img, decoded)
    autoencoder.compile(loss = keras.losses.mean_squared_error,
                        optimizer = keras.optimizers.Adam())
```

[1] J. Cogan, M. Kagan, E. Strauss, and A. Schwartzman, Jet-images: Computer vision inspired techniques for jet tagging, J. High Energy Phys. 02 (2015) 118.

[2] L. G. Almeida, M. Backovic, M. Cliche, S. J. Lee, and M. Perelstein, Playing tag with ANN: Boosted top identification with pattern recognition, J. High Energy Phys. 07 (2015) 086.

[3] L. de Oliveira, M. Kagan, L. Mackey, B. Nachman, and A. Schwartzman, Jet-images–Deep learning edition, J. High Energy Phys. 07 (2016) 069.

[4] 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**, 094034 (2016).

[5] J. Barnard, E. N. Dawe, M. J. Dolan, and N. Rajcic, Parton shower uncertainties in jet substructure analyses with deep neural networks, Phys. Rev. D **95**, 014018 (2017).

[6] G. Kasieczka, T. Plehn, M. Russell, and T. Schell, Deep-learning top taggers or the end of QCD?, J. High Energy Phys. 05 (2017) 006.

[7] G. Louppe, K. Cho, C. Becot, and K. Cranmer, QCD-aware recursive neural networks for jet physics, J. High Energy Phys. 01 (2019) 057.

[8] J. Pearkes, W. Fedorko, A. Lister, and C. Gay, Jet constituents for deep neural network based top quark tagging, arXiv:1704.02124.

[9] K. Datta and A. Larkoski, How much information is in a jet?, J. High Energy Phys. 06 (2017) 073.

[10] A. Butter, G. Kasieczka, T. Plehn, and M. Russell, Deep-learned top tagging with a Lorentz layer, SciPost Phys. **5**, 028 (2018).

[11] S. Egan, W. Fedorko, A. Lister, J. Pearkes, and C. Gay, Long short-term memory (LSTM) networks with jet constituents for boosted top tagging at the LHC, arXiv:1711.09059.

[12] S. Macaluso and D. Shih, Pulling out all the tops with computer vision and deep learning, J. High Energy Phys. 10 (2018) 121.

[13] J. Guo, J. Li, T. Li, F. Xu, and W. Zhang, Deep learning for the R-parity violating supersymmetry searches at the LHC, Phys. Rev. D **98**, 076017 (2018).

[14] S. Choi, S. J. Lee, and M. Perelstein, Infrared safety of a neural-net top tagging algorithm, J. High Energy Phys. 02 (2019) 132.

[15] S. H. Lim and M. M. Nojiri, Spectral analysis of jet substructure with neural network: Boosted Higgs case, J. High Energy Phys. 10 (2018) 181.

[16] P. T. Komiske, E. M. Metodiev, and M. D. Schwartz, Deep learning in color: Towards automated quark/gluon jet discrimination, J. High Energy Phys. 01 (2017) 110.

[17] T. Cheng, Recursive neural networks in quark/gluon tagging, Comput. Software Big Sci. **2**, 3 (2018).

[18] H. Luo, M.-x. Luo, K. Wang, T. Xu, and G. Zhu, Quark jet versus gluon jet: Deep neural networks with high-level features, Sci. China Phys. Mech. Astron. **62**, 991011 (2019).

[19] W. Bhimji, S. A. Farrell, T. Kurth, M. Paganini, Prabhat, and E. Racah, Deep neural networks for physics analysis on low-level whole-detector data at the LHC, J. Phys. Conf. Ser. **1085**, 042034 (2018).

[20] T. Q. Nguyen, D. Weitekamp, D. Anderson, R. Castello, O. Cerri, M. Pierini, M. Spiropulu, and J.-R. Vlimant, Topology classification with deep learning to improve real-time event selection at the LHC, Comput. Software Big Sci. **3**, 12 (2019).

[21] M. Abdughani, J. Ren, L. Wu, and J. M. Yang, Probing stop with graph neural network at the LHC, J. High Energy Phys. 08 (2019) 055.

[22] L. M. Dery, B. Nachman, F. Rubbo, and A. Schwartzman, Weakly supervised classification in high energy physics, J. High Energy Phys. 05 (2017) 145.

[23] T. Cohen, M. Freytsis, and B. Ostdiek, (Machine) learning to do more with less, J. High Energy Phys. 02 (2018) 034.

[24] E. M. Metodiev, B. Nachman, and J. Thaler, Classification without labels: Learning from mixed samples in high energy physics, J. High Energy Phys. 10 (2017) 174.

[25] J. A. Aguilar-Saavedra, J. H. Collins, and R. K. Mishra, A generic anti-QCD jet tagger, J. High Energy Phys. 11 (2017) 163.

[26] E. M. Metodiev and J. Thaler, Jet Topics: Disentangling Quarks and Gluons at Colliders, Phys. Rev. Lett. **120**, 241602 (2018).

[27] A. Andreassen, I. Feige, C. Frye, and M. D. Schwartz, JUNIPR: A framework for unsupervised machine learning in particle physics, Eur. Phys. J. C **79**, 102 (2019).

[28] J. H. Collins, K. Howe, and B. Nachman, CWoLa Hunting: Extending the Bump Hunt with Machine Learning, Phys. Rev. Lett. **121**, 241803 (2018).

[29] R. T. D'Agnolo and A. Wulzer, Learning new physics from a machine, Phys. Rev. D **99**, 015014 (2019).

[30] J. W. Monk, Deep learning as a Parton shower, J. High Energy Phys. 12 (2018) 021.

[31] A. De Simone and T. Jacques, Guiding new physics searches with unsupervised learning, Eur. Phys. J. C **79**, 289 (2019).

[32] J. Hajer, Y.-Y. Li, T. Liu, and H. Wang, Novelty detection meets collider physics, arXiv:1807.10261.

[33] A. A. Pol, G. Cerminara, C. Germain, M. Pierini, and A. Seth, Detector monitoring with artificial neural networks at the CMS experiment at the CERN Large Hadron Collider, Comput. Software Big Sci. **3**, 3 (2019).

[34] P. T. Komiske, E. M. Metodiev, B. Nachman, and M. D. Schwartz, Pileup mitigation with machine learning (PUMML), J. High Energy Phys. 12 (2017) 051.

[35] L. de Oliveira, M. Paganini, and B. Nachman, Learning particle physics by example: Location-aware generative adversarial networks for physics synthesis, Comput. Software Big Sci. **1**, 4 (2017).

[36] D. Guest, K. Cranmer, and D. Whiteson, Deep learning and its application to LHC physics, Annu. Rev. Nucl. Part. Sci. **68**, 161 (2018).

[37] P. Baldi and K. Hornik, Neural networks and principal component analysis: Learning from examples without local minima, Neural Netw. **2**, 53 (1989).

[38] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in *ICML* (2008), https://doi.org/10.1145/1390156.1390294.

[39] D. P. Kingma and M. Welling, Auto-encoding variational bayes, arXiv:1312.6114.

[40] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT, Cambridge, MA, 2016).

[41] B. R. Kiran, D. M. Thomas, and R. Parakkal, An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos, arXiv:1801.03149.

[42] Building Autoencoders in Keras, https://blog.keras.io/building-autoencoders-in-keras.html.

[43] T. Sjostrand, S. Ask, J. R. Christiansen, R. Corke, N. Desai, P. Ilten, S. Mrenna, S. Prestel, C. O. Rasmussen, and P. Z. Skands, An introduction to PYTHIA 8.2, Comput. Phys. Commun. **191**, 159 (2015).

[44] J. de Favereau, C. Delaere, P. Demin, A. Giammanco, V. Lemaitre, A. Mertens, and M. Selvaggi (DELPHES 3 Collaboration), DELPHES 3, A modular framework for fast simulation of a generic collider experiment, J. High Energy Phys. 02 (2014) 057.

[45] M. Cacciari, G. P. Salam, and G. Soyez, FastJet user manual, Eur. Phys. J. C **72**, 1896 (2012).

[46] T. Heimel, G. Kasieczka, T. Plehn, and J. M. Thompson, QCD or what?, SciPost Phys. **6**, 030 (2019).

[47] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, Generative adversarial networks, arXiv:1406.2661.

[48] A. Makhzani, J. Shlens, N. Jaitly, and I. J. Goodfellow, Adversarial autoencoders, arXiv:1511.05644.

[49] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, Unsupervised anomaly detection with generative adversarial networks to guide marker discovery, arXiv:1703.05921.

[50] S. Chang, T. Cohen, and B. Ostdiek, What is the machine learning?, Phys. Rev. D **97**, 056009 (2018).