

A Jet-Multiplicity Confound in Reconstruction-Based Anomaly Detection: Evidence from a Higgs-to-Tau-Pair Proxy Signal

Abhijay Arun

GEMS Modern Academy

Nad Al Sheba, Dubai, United Arab Emirates

abhijay.a.arun@gmail.com

Abstract

Unsupervised anomaly detectors trained on background events alone have been proposed for model-independent searches at the Large Hadron Collider. This study tests, for one proxy signal, whether a model that reconstructs the background better also detects an unseen signal better. Principal component analysis (PCA), an autoencoder (AE) and a variational autoencoder (VAE) with latent sizes from 2 to 16 are trained on simulated background events from the ATLAS Higgs Boson Machine Learning Challenge 2014 dataset, and their reconstruction error ranks Higgs boson decays to tau pairs, which are never used in training or tuning. Raw scores reach an area under the receiver operating characteristic curve (AUC) of 0.55 to 0.61, but the jet count alone, used as a score, reaches 0.59: signal events have more jets, and reconstruction error rises with jet count. Ranked only against background events with the same jet count, the AE and VAE fall to or below chance (mean AUC 0.47 to 0.50). Only PCA reconstruction with two or four components stays above chance, and its larger edge (1.61% of signal at a 1% false-positive rate, against 1% for chance) is reproduced on a held-out half of the signal. Below-chance scores trace back to signal taus and leptons being more central in pseudorapidity. Simple baselines, scores on the latent codes, a masked score and retraining without jet inputs leave the picture unchanged, and the detectors that catch independent feature shifts miss correlated energy-scale shifts. Better reconstruction never brought detection clearly above chance.

Keywords

Computer Science and Software Engineering, Machine Learning, Anomaly Detection, Autoencoders, Particle Physics

ACM Reference Format:

Abhijay Arun. 2026. A Jet-Multiplicity Confound in Reconstruction-Based Anomaly Detection: Evidence from a Higgs-to-Tau-Pair Proxy Signal. In *Proceedings of International Journal of Secondary Computing and Applications Research (IJSCAR VOL. 3, ISSUE 4)*. ACM, New York, NY, USA, 16 pages. <https://doi.org/10.67149/yhjs2024.5/6uf9c50f>

1 Introduction

In 2012 the ATLAS and CMS experiments at the Large Hadron Collider (LHC) reported the observation of a new boson with a mass near 125 GeV, compatible with the Standard Model Higgs

boson [5, 13]. Collins, Howe and Nachman noted that searches at the LHC had found no significant evidence for physics beyond the Standard Model, and argued that the search program needs wider sensitivity to unexpected scenarios [14]. Community challenges such as the LHC Olympics 2020 and the Dark Machines Anomaly Score Challenge were set up to develop and compare model-agnostic anomaly detection methods on simulated collider data [1, 22], and a recent review surveys the machine learning techniques now used for anomaly detection in particle physics [10].

A common unsupervised approach trains an autoencoder on background events only. The network learns to compress and rebuild those events, and an event that it rebuilds poorly is treated as anomalous [16]. Autoencoders of this kind have been used to search for anomalies in jet substructure [20], variational autoencoders have been trained on known processes to flag events from unseen processes as outliers [12], and autoencoders have been implemented on field-programmable gate arrays (FPGAs) for real-time event selection at the LHC [19].

The dataset used in this study comes from the Higgs Boson Machine Learning Challenge, which was organized to promote collaboration and used simulated data that ATLAS physicists had used in a search for the Higgs boson [2]. Its signal process, the decay of a Higgs boson to a pair of tau leptons, has been studied by ATLAS in collision data, where the evidence for the coupling of the Higgs boson to tau leptons reached an observed significance of 4.5 standard deviations [7]. Supervised deep networks trained with labels improve on earlier approaches in benchmark searches for exotic particles [8] and for Higgs boson decays to tau pairs [9].

Anomaly detection methods differ in how much they assume about the signal. Unsupervised methods train on background-like data only and score how unusual each event is, as in the autoencoder studies above [12, 16, 20]. Weakly supervised methods instead train a classifier on mixed samples. In classification without labels (CWoLa), a classifier is trained to distinguish statistical mixtures of classes, and neither individual labels nor class proportions are required [29]. Collins, Howe and Nachman built a resonance search on this idea that only requires the signal to be localized in one known direction of phase space [14]. Table A1 in the Appendix summarizes the studies most relevant here.

The approach rests on an assumption: the events a model rebuilds badly are the events of interest. Several studies question it. Finke *et al.* showed that an autoencoder that tags top jets against a background of quantum chromodynamics (QCD) jets performs poorly when the roles are reversed, and concluded that the standard autoencoder setup cannot be considered a model-independent anomaly tagger [17]. Fraser *et al.* found that hyperparameter choices strongly affect performance, that the best settings for one signal are

This paper is published under the Creative Commons Attribution-NonCommercial-NoDerivs 4.0 International (CC-BY-NC-ND 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.

IJSCAR VOL. 3, ISSUE 4,
© 2026 The Author(s). Published by IJSCAR.
<https://doi.org/10.67149/yhjs2024.5/6uf9c50f>

not the best for another, and that the choices that best represent the background are not necessarily best for signal identification [18]. Outside physics, deep generative models trained on one image dataset have assigned higher likelihood to images from a different dataset [30]. A related problem is a score that depends on a nuisance quantity. For classifiers, adversarial training can make the output independent of nuisance parameters [27], Heime *et al.* used adversarial mass decorrelation for an autoencoder tagger [20], and Kasieczka and Shih introduced a distance-correlation penalty that matches the performance of adversarial decorrelation while being simpler and more stable to train [23]. An anomaly score that follows a simple event property, rather than the signal, raises the same concern. For variational autoencoders, Burda, Grosse and Salakhutdinov call a latent dimension active when the variance of its posterior mean across inputs exceeds 10^{-2} ; they found that these models use far fewer active dimensions than their capacity allows, and that adding latent dimensions did not increase the number of active dimensions [11]. To ask what learned codes contain, linear classifier probes can be trained separately from a model on its intermediate features to measure how useful those features are for a classification task [3].

This study asks whether a model that reconstructs the background better also detects an unseen signal better, and what the detectors respond to instead. Finke *et al.* and Fraser *et al.* already showed that the standard autoencoder setup is not model independent and that the best background representation is not necessarily the best for signals [17, 18], so the contribution here is narrower. Several earlier studies work with individual jets [16–18, 20], whereas this study uses event-level features from the ATLAS Higgs Boson Machine Learning Challenge 2014 dataset [2, 6], with Higgs boson decays to tau pairs as a proxy signal that the models never see in training. Principal component analysis (PCA), a fully connected autoencoder (AE) and a variational autoencoder (VAE) are compared at four latent sizes, with three training seeds for each neural model, under one fixed protocol. The main finding is a confound: most of the apparent detection follows jet multiplicity rather than the signal. A jet-count control is checked on a held-out half of the signal and against simple baselines, alternative scores and alternative input encodings, and synthetic shifts and linear probes separate what the detectors respond to from what their codes contain. The proxy signal has the same final state as one background component, Z boson decays to tau pairs, so the results describe one hard case, not model-independent detection in general.

2 Methods

2.1 Dataset

The dataset comes from the ATLAS Higgs Boson Machine Learning Challenge 2014 [6] and is released through the CERN Open Data Portal under a CC0 license. It holds 818,238 simulated collision events produced with the official ATLAS full detector simulation. Every event contains exactly one electron or muon and one hadronic tau [2]. The signal class is Higgs boson decays to tau pairs (Higgs mass 125 GeV), and the background class is a mixture of Z boson decays to tau pairs, top-quark pair production and W boson decays [6].

In this study the labeled Higgs events act as a stand-in for an unexpected process. These events are called the *proxy signal*. Signal labels are used only for evaluation and for labeled reference probes (Methods, Supervised reference and latent probes), never for training or tuning. The study makes no claim about new physics.

2.2 Features and missing values

Each event is described by 30 features [2]. Thirteen are derived quantities (prefix DER), computed from the primitive features, and seventeen are primitive quantities (prefix PRI), which are raw measurements of the collision made by the detector [2]. Features that cannot be computed for an event are set to -999.0 , a value outside the normal range of every variable. The jet count PRI_jet_num takes the values 0, 1, 2 or 3, with larger counts capped at 3. The leading-jet features are undefined when PRI_jet_num is 0, and the subleading-jet and jet-pair features are undefined when it is 1 or less [2].

2.3 Preprocessing

- (1) Replace -999.0 with a missing-value marker.
- (2) Record three binary indicators: fewer than two jets (DER_mass_jet_jet missing), zero jets (PRI_jet_leading_pt missing), and missing DER_mass_MMC.
- (3) Fill each missing value with the median of that feature over the training background.
- (4) Apply the scikit-learn QuantileTransformer [32] with 1000 quantiles, fitted on the training background only. It maps each feature through its estimated cumulative distribution function and then the normal quantile function, and values outside the fitted range go to the bounds of the output distribution [34].
- (5) Append the three indicators after the transform.

Each event is then a vector with $d = 33$ entries. Signal events never reach the imputer or the transform.

2.4 Data splits

The splits are made with a fixed random seed (0) and listed in Table A2 in the Appendix. Background events are divided 70%, 15% and 15% into training (377,074 events), validation (80,802) and test (80,802) sets. Signal events are divided into two halves of 139,780. One half (sig_clf) is reserved for the supervised reference and the probes. The other half (sig_eval) is used only for evaluation. The preprocessing script checks that no signal event appears in the training, validation or test sets.

2.5 Reconstruction-error anomaly scores

Each model learns to reconstruct training background events through a low-dimensional latent code of size $k \in \{2, 4, 8, 16\}$. The anomaly score of an event $\mathbf{x}$ with reconstruction $\hat{\mathbf{x}}$ is its mean squared reconstruction error,

$$s(\mathbf{x}) = \frac{1}{d} \sum_{j=1}^d (x_j - \hat{x}_j)^2, \quad (1)$$

so a higher score means more anomalous.

PCA. Principal component analysis [35] is a linear baseline. It is fitted on the training background with scikit-learn [32]. Each event is reconstructed by projecting it onto k components and mapping back. PCA is fitted once for each k .

Autoencoder (AE). An autoencoder compresses its input into a low-dimensional code and reconstructs the input from that code [21]. The network used here is fully connected. The encoder has layers of size $d \rightarrow 64 \rightarrow 32 \rightarrow k$, the decoder mirrors it, hidden layers use ReLU, and the output layer is linear. The training loss is the squared error summed over features and averaged over the batch.

Variational autoencoder (VAE). The VAE [25] shares the encoder body of the AE but has two linear heads that output a mean μ and a log-variance $\log \sigma^2$ for the latent code. During training a code is sampled with the reparameterization trick. The loss is

$$\mathcal{L} = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 + \beta D_{\text{KL}}(q(\mathbf{z} | \mathbf{x}) \| \mathcal{N}(\mathbf{0}, \mathbf{I})), \quad (2)$$

averaged over the batch, with $\beta = 1$. At scoring time the decoder is fed the mean code μ with no sampling.

2.6 Training

Both neural models are trained with Adam [24] in PyTorch [31], using a learning rate of 10^{-3} , a batch size of 512 and no weight decay. Training stops early when the validation background loss has not improved for 15 epochs, and the weights from the best epoch are restored. Training is capped at 300 epochs, and 9 of the 24 neural runs reached the cap. Three seeds (0, 1 and 2) are trained for each model and each k , which gives 24 neural runs.

2.7 Evaluation metrics

Scores of test background events (label 0) and sig_eval events (label 1) are compared with two measures. The first is the area under the ROC curve (AUC), where 0.5 is chance. The second is signal efficiency at a fixed false-positive rate α . A threshold is placed so that a fraction α of test background events score above it, and the efficiency is the fraction of sig_eval events above the same threshold. It is read from the ROC curve by linear interpolation. The primary metric is signal efficiency at a 1% false-positive rate, for which chance is 1%. It is also reported at 0.1%. A configuration is called above chance only when the 95% intervals of both its AUC and its efficiency lie above the chance values. No setting or direction was chosen using signal performance.

2.8 Control for jet multiplicity

Many features depend on the number of jets, so raw reconstruction errors can differ between jet-count groups. As a control, each score is also converted to a percentile among validation background events with the same jet count. For an event with jet count j ,

$$p(\mathbf{x}) = \frac{|\{\mathbf{v} \in V_j : s(\mathbf{v}) \leq s(\mathbf{x})\}|}{|V_j|}, \quad (3)$$

where V_j is the set of validation background events with jet count j . The percentiles of all events are then pooled and evaluated. This

separates the jet-count effect from the signal. Metrics are also reported inside each jet-count group. This control was added after the first results were seen, so it is a post hoc analysis.

2.9 Synthetic shift study

A synthetic shift study tests how the detectors respond to controlled changes. The test background is split in half with a fixed seed. One half is the unshifted reference. In the other half, a constant $s \in \{0, 0.25, 0.5, 1, 2\}$ is added to a chosen set of features in the scaled space, where each feature has a normal shape, so s is measured in standard deviations. The shifted features are drawn at random from the 18 features that are always defined (excluding PRI_jet_num). Sets of 1, 3 and 10 features are used (three random sets for each size), along with the full set of 18. Shifted events play the role of the signal class when computing the metrics. The case $s = 0$ is a null check, since both halves then come from the same distribution.

2.10 Supervised reference and latent probes

As an upper reference, a supervised classifier is trained on the training background and 80% of sig_clf, with the remaining 20% used for validation. It is a fully connected network ($d \rightarrow 64 \rightarrow 32 \rightarrow 1$, ReLU) trained with binary cross-entropy and Adam (learning rate 10^{-3} , batch size 512), with early stopping after 10 epochs without improvement. It is scored on the test background and sig_eval.

To ask what the latent codes contain, linear probes [3] are fitted on the validation background (and sig_clf for the signal probe) and scored on the test set. Ridge regression predicts each always-defined feature from the code, and R^2 is reported. Logistic regression predicts the jet count and, as a labeled reference, the signal class. For VAEs, a latent dimension is called active when the variance of μ across events is above 0.01, following Burda, Grosse and Salakhutdinov [11]. For visual inspection, codes of 5,000 background and 5,000 signal events from the seed-0 runs are embedded with PCA and uniform manifold approximation and projection (UMAP) [28].

2.11 Further checks

Six further checks reuse the same splits, with every fit made on training background only: a confirmation of the jet-corrected evaluation on sig_clf, which the detectors never saw and which played no part in choosing the correction; simple baselines; unsupervised scores on the latent codes; a score masked to the features defined for each event; retraining without direct jet inputs; and correlated energy-scale shifts. They are described in the Appendix.

2.12 Uncertainty

For PCA, which is fitted once, the test scores are resampled with a multinomial bootstrap over score bins [15]. This uses 1000 resamples, and 95% percentile intervals are reported. A wider interval, from the 0.625th to the 99.375th percentile, allows for comparing four PCA sizes at once. For the AE and VAE, the mean and standard deviation over the three seeds are reported. Rank correlations are Spearman coefficients computed with SciPy [36]. Because runs that share a configuration are not independent, correlations are reported both over runs and over configurations with seeds averaged, and separately for PCA and the neural models.

2.13 Software and reproducibility

All code is written in Python 3.11 and uses PyTorch 2.14.1, scikit-learn 1.9.1, SciPy 1.17.1, NumPy 2.4.6, pandas 3.0.6, matplotlib 3.11.2 and umap-learn 0.5.12, with these versions pinned in the repository's requirements file. Every split and model run uses a fixed seed, and repeated runs give identical results. The code, results and figures are archived on Zenodo [4].

3 Results and Discussion

3.1 Reconstruction quality and detection

All three models rebuild background events better as the latent size grows. The validation reconstruction error, the squared error summed over the 33 features, falls from 24.0 to 4.5 for PCA between $k = 2$ and $k = 16$, and from 12.9 to 0.74 for the AE. For the VAE it falls from 13.8 to 5.7 at $k = 8$ and then stays flat (5.9 at $k = 16$). Figure 1 plots this error against the AUC on the proxy signal for all 28 runs. Pooled over all runs, the Spearman rank correlation between reconstruction error and AUC is -0.061 for the raw score and -0.061 for the jet-corrected score ($n = 28$, $p = 0.76$; the two agree to three decimals by coincidence). Over the 12 configurations with seeds averaged it is $+0.29$ ($p = 0.35$). The pooled numbers hide opposite trends. For PCA, the corrected AUC falls as components are added beyond four ($\rho = 0.80$ between error and AUC over the four sizes, $p = 0.20$). For the eight AE and VAE configurations, the corrected AUC rises as the error falls ($\rho = -0.76$, $p = 0.03$), but only from 0.472 to 0.504, that is, from below chance toward chance.

3.2 Jet count as a confound

Signal and background events have different jet-count mixes (Table 1). Background is dominated by zero-jet events (45.2%), while signal has 29.7% zero-jet events and about twice as many two-jet events as background (30.0% against 15.1%). The mean raw score of a trained AE ($k = 8$, seed 0) rises with jet count in both classes, from 0.062 for zero-jet background events to 0.221 for three-jet background events. Within each jet count the mean scores of the two classes differ by at most 0.016, and the AUC of that model inside a single jet count lies between 0.465 and 0.536. A score that tracks jet count can therefore separate the two classes without detecting anything beyond a single simple feature. The jet count itself, used as a score, reaches a raw AUC of 0.587, inside the range of the reconstruction-based scores. An isolation forest and the nearest-neighbour distance behave in the same way: their raw AUCs of 0.562 and 0.594 fall to 0.491 and 0.491 after the correction (Table A3 in the Appendix). Table 2 compares the raw score with the percentile-within-jet-count score (Methods, Control for jet multiplicity), and Figure A1 in the Appendix shows both across latent sizes.

The raw score looks better than chance in every configuration, with a mean AUC between 0.547 (VAE, $k = 4$) and 0.615 (PCA, $k = 4$). The correction lowers the mean AUC in all 12 configurations, by 0.06 to 0.10. Raw efficiency at 1% false-positive rate is already close to chance, between 0.86% and 1.74% across configurations.

After the correction, only PCA at $k = 2$ and $k = 4$ stays above chance on both measures, and $k = 4$ is the stronger. Its AUC is 0.553 (95% bootstrap interval 0.550 to 0.555), and its efficiency at 1% false-positive rate is 1.61% (1.51% to 1.74%), against 1.00% for

chance. At $k = 2$ the AUC is 0.525 (0.522 to 0.527) and the efficiency 1.15% (1.07% to 1.28%). At $k = 8$ the AUC interval (0.505 to 0.510) lies above 0.5, but the efficiency, 0.93%, does not exceed chance. At $k = 16$ the AUC is 0.460 (0.457 to 0.462), below chance. On the confirmation sample sig_clf, PCA at $k = 4$ gives an AUC of 0.552 (0.549 to 0.554) and an efficiency of 1.59% (1.49% to 1.71%), and PCA at $k = 2$ gives 0.524 and 1.17% (1.08% to 1.29%), so both edges are reproduced.

The corrected mean AUC of the AE and VAE lies between 0.472 and 0.504 across their eight configurations. In 19 of the 24 runs, the 95% bootstrap interval lies entirely below 0.5 (20 of 24 on sig_clf). Three runs lie entirely above it and two include it. In five configurations (AE at $k = 2$ and 4, VAE at $k = 2$, 4 and 8) all three seeds fall below 0.5. Only the AE at $k = 16$ shows an efficiency above chance in some runs: seeds 1 and 2 reach 1.45% and 1.41% and seed 0 reaches 0.74%, which gives a mean of 1.20% with a standard deviation of 0.40 percentage points.

3.3 Why many scores fall below chance

The below-chance AUCs are not noise, but the pattern is mixed across jet counts. Inside three-jet events every configuration lies between 0.446 and 0.484, and PCA at $k = 16$ reaches 0.410 inside zero-jet events, while in zero-jet events the AE at $k = 8$ (seed 0) is at 0.536 (Table 1). Where scores fall below chance, signal events are rebuilt slightly better than background events with the same jet count. The pseudorapidities of the tau and the lepton drive this. For the AE at $k = 8$ (seed 0), PRI_tau_eta and PRI_lep_eta show the largest drop in squared error for signal relative to background in every jet-count group, and the mean error on PRI_tau_eta is 0.149 for signal against 0.289 for background. Signal taus and leptons are more central: in zero-jet events the mean $|\eta|$ is 0.93 for signal and 1.07 for background taus, and 0.97 against 1.17 for leptons, with similar gaps for one, two and three jets. Central values sit near the middle of the scaled feature, and the AE's error on PRI_tau_eta grows with the squared scaled value (Pearson correlation 0.39 on test background). For PCA at $k = 16$ in events with two or three jets, the jet-pair pseudorapidity features DER_deltaeta_jet_jet and DER_prodelta_jet_jet play the same role. Signal events are not closer to the centre of the scaled feature space overall (the AUC of the mean squared scaled value is 0.505 to 0.553 per jet group), so the effect comes from a few angular features.

For reference, the supervised network trained with signal labels (Methods, Supervised reference and latent probes) reaches an AUC of 0.911 and an efficiency of 33.7% at 1% false-positive rate on the same test sets. The proxy signal can therefore be separated with labels, while reconstruction error does not use that information.

3.4 Synthetic shifts

The null case $s = 0$ gives chance-level detection (mean AUC 0.498, range 0.496 to 0.500; mean efficiency 0.96%).

Shifted events are detected far more easily than the proxy signal (Table A4 and Figures A2 and A3 in the Appendix). When all 18 eligible features move by 2 standard deviations, the AE at $k = 8$ and $k = 16$ catches more than 99.9% of shifted events at 1% false-positive rate, and the VAE at $k = 8$ catches 99.6%. PCA catches between 48.7% ($k = 16$) and 77.5% ($k = 2$).

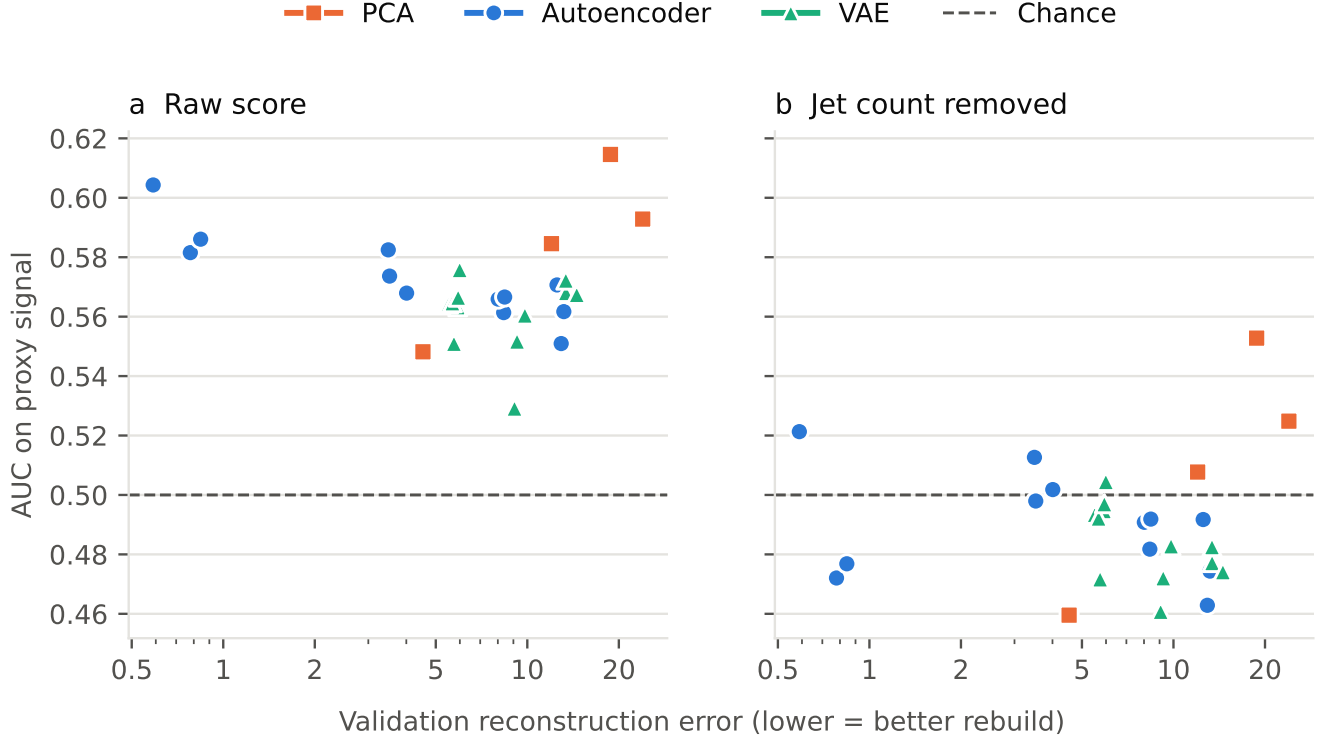


Figure 1: Reconstruction error versus detection. Lower reconstruction error does not bring better detection of the proxy signal.

Note. Each point is one trained model (28 in total; PCA squares, autoencoder circles, variational autoencoder triangles), placed by its validation reconstruction error (log scale, lower means a better rebuild) and its AUC (area under the receiver operating characteristic curve) from (a) the raw score or (b) the score converted to a percentile within its jet count; the dashed line marks chance (AUC 0.5). Created with Python 3.11 and matplotlib 3.11.2.

Table 1: Jet-count mix and raw scores by jet count. Signal events have more jets than background, and the raw score of a trained autoencoder rises with jet count in both classes.

Jets	Background (%)	Signal (%)	Mean score, background	Mean score, signal	AUC within jet count
0	45.2	29.7	0.062	0.066	0.536
1	30.2	32.3	0.134	0.126	0.474
2	15.1	30.0	0.211	0.211	0.506
3	9.5	8.1	0.221	0.206	0.465

Note. Shares are for test background and sig_eval signal; the mean score is the mean squared reconstruction error of the autoencoder with latent size $k = 8$ (seed 0); AUC (area under the receiver operating characteristic curve) is computed inside each jet count. Computed with Python 3.11 and scikit-learn 1.9.1.

Yet after the jet-count correction the same two AE configurations detect at most 1.5% of the proxy signal in any single run at the same false-positive rate (Table 2).

Correlated energy-scale shifts are almost invisible. When all 14 energy-like features are multiplied by a common factor of up to 1.20, the raw AUC of every run lies between 0.488 and 0.525, and at most 1.22% of shifted events are caught at a 1% false-positive rate (1.45% after the jet-count correction). Scaling only the jet features gives raw AUCs between 0.491 and 0.505 (Table A5 in the Appendix). The high detection rates above therefore come from moving events off the background distribution, not from sensitivity to changes that keep the correlations between features.

3.5 What the latent codes contain

Figure A4 and Table A6 in the Appendix summarize the probes described in the Methods, and Figures A5 to A7 give further detail. Jet count can be read from the codes with 93.7% to 100% accuracy, against 45.2% for always guessing the most common jet count. From $k = 8$ the AE and VAE codes give 100.0% accuracy, and PCA gives 99.6% at $k = 8$ and 99.8% at $k = 16$.

The proxy signal is also present in the codes. The AUC within jet count of the linear signal probe rises with latent size, from 0.586 to 0.607 at $k = 2$ to 0.747 to 0.762 at $k = 16$, against 0.801 for the same probe on the raw inputs. At $k = 16$ the probe's efficiency at 1% false-positive rate is 6.0% for PCA, 5.9% for the AE and 7.7% for the VAE, compared with 9.5% on the raw inputs, at best 1.6% for

Table 2: Detection of the proxy signal before and after the jet-count correction. The jet-count correction lowers the AUC of every configuration by 0.06 to 0.10, leaving only PCA at $k = 2$ and $k = 4$ above chance.

Model	k	Recon. error	Raw AUC	Raw eff. (%)	Corrected AUC	Corrected eff. (%)
PCA	2	23.96	0.593	1.01	0.525	1.15
PCA	4	18.77	0.615	1.59	0.553	1.61
PCA	8	12.01	0.585	1.74	0.508	0.93
PCA	16	4.54	0.548	1.25	0.460	0.66
AE	2	12.88	0.561 ± 0.010	0.90 ± 0.06	0.476 ± 0.015	0.83 ± 0.06
AE	4	8.27	0.565 ± 0.003	1.10 ± 0.06	0.488 ± 0.006	0.80 ± 0.08
AE	8	3.67	0.575 ± 0.007	1.37 ± 0.27	0.504 ± 0.008	0.87 ± 0.05
AE	16	0.74	0.591 ± 0.012	1.10 ± 0.43	0.490 ± 0.027	1.20 ± 0.40
VAE	2	13.78	0.569 ± 0.003	0.86 ± 0.05	0.478 ± 0.004	0.84 ± 0.04
VAE	4	9.38	0.547 ± 0.016	0.94 ± 0.26	0.472 ± 0.011	0.83 ± 0.09
VAE	8	5.69	0.564 ± 0.001	1.34 ± 0.19	0.493 ± 0.001	0.70 ± 0.09
VAE	16	5.88	0.564 ± 0.013	1.07 ± 0.06	0.491 ± 0.017	0.65 ± 0.12

Note. k , latent size; Recon. error, validation reconstruction error (squared error summed over the 33 features, averaged over validation background events); AUC, area under the receiver operating characteristic curve; eff., signal efficiency at a 1% false-positive rate; PCA, principal component analysis; AE, autoencoder; VAE, variational autoencoder. Values are means $\pm$ standard deviation over three seeds; PCA is a single fit, so no spread is given. Chance is an AUC of 0.5 and an efficiency of 1.00%. Computed with Python 3.11, PyTorch 2.14.1 and scikit-learn 1.9.1.

reconstruction error, and 33.7% for the supervised network. The VAE uses 2.0, 4.0, 7.3 and 7.0 active latent dimensions at $k = 2, 4, 8$ and 16.

Unsupervised scores on the same codes recover little of this. After the jet-count correction, the best are the Mahalanobis distance of the VAE code at $k = 16$ (AUC 0.552 ± 0.016 , efficiency $1.53\% \pm 0.80\%$), the Mahalanobis distance of the PCA code at $k = 16$ (0.549, interval 0.546 to 0.552; 1.85%, 1.74% to 1.99%) and the Gaussian-mixture likelihood of the PCA code at $k = 4$ (0.555; 1.11%, 1.02% to 1.21%). Both PCA scores meet the above-chance definition, but neither exceeds 2%. The negative evidence lower bound of the VAE behaves like its reconstruction error, with corrected AUCs of 0.487 to 0.502 (Table A3).

3.6 Better reconstruction is not better detection

Better reconstruction never brought detection clearly above chance. For the AE and VAE it moved the corrected AUC from below chance toward chance, and for PCA, components beyond four made detection worse. The AE at $k = 16$ rebuilds background events with an error of 0.74, about 25 times lower than PCA at $k = 4$ (18.77), yet PCA at $k = 4$ is the better detector of the proxy signal after the jet-count correction (AUC 0.553 against 0.490). This agrees with the earlier findings of Fraser *et al.* [18] and Finke *et al.* [17] described in the Introduction. The VAE settles at about seven active dimensions at both $k = 8$ and $k = 16$ (7.3 and 7.0), in line with the observation of Burda, Grosse and Salakhutdinov that adding latent dimensions did not increase the number of active dimensions [11]. For the VAE, mean squared error is only part of how well the model describes the background, and its plateau at 5.7 to 5.9 partly reflects the KL term with $\beta = 1$. Scoring with the full negative evidence lower bound instead of the squared error does not change its detection, though.

3.7 Shifted events and the proxy signal

The detectors respond to anomalies of one kind. Independent shifts in the scaled space move events off the background distribution, which is what a reconstruction-based detector is built to flag, and they are caught far more often than the proxy signal (Table A4).

Energy-scale shifts that keep the correlations between features are not caught, and neither is the proxy signal, whose events receive reconstruction errors close to those of background events with the same jet count (Table 1). Here, being poorly rebuilt and being signal-like turned out to be different properties.

3.8 Jet multiplicity as a confound

The raw scores gave a mean AUC between 0.55 and 0.61 for every configuration, which on its own would read as a modest success. Jet multiplicity is a confound: the two classes have different jet-count mixes, and the score rises with jet count (Table 1). The jet count alone reaches a raw AUC of 0.587, so a score that follows it ranks signal above background without detecting anything beyond a single simple feature. When each score is compared only with background events of the same jet count, the apparent detection largely disappears, for the reconstruction scores and for the isolation forest and nearest-neighbour baselines alike.

The confound is not only a product of the input encoding. Masking undefined features weakens the rise of the score with jet count but does not remove it, and the mean background score of the AE at $k = 8$ (seed 0) still grows from 0.103 with zero jets to 0.241 with three. Retraining without PRI_jet_num and the two jet indicators changes little. For the AE at $k = 8$ (means over three seeds), jet count can still be read from the code with 92.7% accuracy, the mean background score still grows from 0.063 with zero jets to 0.230 with three, and the corrected AUC is 0.498 ± 0.013 (Table A3).

Metrics for unsupervised detectors should therefore also be reported within groups of any event property that the score tracks and whose distribution differs between signal and background. The percentile correction used here only repairs the evaluation after training. Methods built to make a classifier independent of a nuisance variable, adversarial decorrelation [20, 27] or a distance-correlation penalty [23], could be applied to the anomaly score and the jet count during training. That was not tested here.

3.9 What the codes contain and what the scores use

The label-trained probe reaches 5.9% to 7.7% efficiency on the codes at $k = 16$, and the unsupervised scores at most 1.85%. Because the probe uses labels, this gap shows that signal information is present in the codes, not that an unsupervised score could recover it, and neither reconstruction nor density in the learned space separates this signal in a useful way. Weakly supervised training such as CWoLa [14, 29], which uses mixed samples instead of labels, is one option that this study did not test.

3.10 Limitations

Training budget. Nine of the 24 neural runs (five AE and four VAE) reached the 300-epoch cap before early stopping, so they may not have fully converged; longer budgets were not tested.

Post hoc jet-count analysis. The jet-count confound was found after the first results had been seen, so the jet-count correction was not specified in advance. The `sig_clf` confirmation reduces this concern for the PCA edge but uses the same background test set. Models trained separately for each jet count, or trained with a decorrelation penalty, were not tested.

Proxy signal. The proxy signal shares its final state with one background component, Z boson decays to tau pairs, which makes it a hard case for a detector trained on that background. A second, more distinct proxy, such as one background process held out of training, would test the scope of the conclusions, but the public dataset does not record which process produced each background event, so that split could not be made.

Statistical coverage. All splits use one fixed seed, so variation caused by the choice of split is not measured. Only three model seeds were trained per configuration. PCA is a single deterministic fit, and the bootstrap intervals cover noise in the test sample only. For the AE and VAE they are much narrower than the spread across seeds, so the means and standard deviations over seeds are the better summary. The correlations between reconstruction error and AUC rest on four PCA sizes and eight neural configurations, so they are weak evidence on their own.

Scope of the data. All results come from one simulated dataset and one signal process, Higgs boson decays to tau pairs, used as a proxy signal. The conclusions may not carry over to real collision data, to other signals or to other feature sets.

Models and probes. One architecture (hidden layers of 64 and 32 units), one optimizer setting, one VAE weight ($\beta = 1$) and latent sizes from 2 to 16 were used. Other architectures, optimizer settings, values of β and latent sizes were not explored, and the retraining without jet inputs covers only $k = 8$ for the neural models. The latent probes are linear, so information that only a nonlinear readout could extract does not appear in Table A6.

4 Conclusion

This study compared PCA, an autoencoder and a variational autoencoder as unsupervised detectors of one proxy signal, Higgs boson decays to tau pairs, in the ATLAS Higgs challenge data, with

fixed seeds and with signal events kept out of training, validation and tuning. Raw reconstruction error seemed to detect the signal, but the effect came mostly from jet multiplicity: the jet count alone did as well. Within jet-count groups the AE and VAE performed at or below chance, and PCA with two or four components kept a small edge that held on a second half of the signal. The further checks left this picture unchanged, and the latent codes held signal information that the unsupervised scores tested here recovered only in small part.

For future studies, evaluations of unsupervised detectors should report results within groups of any event property, such as jet count, that both the score and the class balance depend on, and should compare against that property used as a score. Natural next steps are decorrelating the score from such properties during training, weakly supervised training, and more distinct proxy signals and datasets.

Declarations and Code Availability

Code and data. The code, result tables and figures are archived on Zenodo [4] and are also available at <https://github.com/Abhijay-MSD/higgs-anomaly-detection>. The script `src/paper_numbers.py` prints the derived numbers quoted in the text, and the README lists the scripts behind the further checks. Neither ATLAS nor CERN endorses this work.

Funding and competing interests. This research received no external funding, and the author declares no competing interests.

Use of AI tools. Claude (Anthropic) was used to debug the analysis code, including scripts for additional checks introduced during revision, and to execute some of these checks. The author conducted the main experiments, reviewed all code and results, verified the reported figures against the underlying result files, and drafted the article. The author takes full responsibility for the content.

Acknowledgments

The author thanks the editor and the three reviewers for their comments. The work was carried out independently, without a research mentor or supervising teacher.

References

- [1] T. Aarrestad, M. van Beekveld, M. Bona, A. Boveia, S. Caron, J. Davies, et al. 2022. The Dark Machines Anomaly Score Challenge: Benchmark Data and Model Independent Event Classification for the Large Hadron Collider. *SciPost Phys.* 12 (2022), 043. doi:10.21468/SciPostPhys.12.1.043
- [2] C. Adam-Bourdarios, G. Cowan, C. Germain, I. Guyon, B. Kégl, and D. Rousseau. 2015. The Higgs Boson Machine Learning Challenge. In *Proceedings of the NIPS 2014 Workshop on High-Energy Physics and Machine Learning (Proceedings of Machine Learning Research, Vol. 42)*. PMLR, 19–55. <https://proceedings.mlr.press/v42/cowa14.html>
- [3] G. Alain and Y. Bengio. 2016. Understanding Intermediate Layers Using Linear Classifier Probes. arXiv preprint arXiv:1610.01644. <https://arxiv.org/abs/1610.01644>
- [4] A. Arun. 2026. A Jet-Multiplicity Confound in Reconstruction-Based Anomaly Detection: Evidence from a Higgs-to-Tau-Pair Proxy Signal. Zenodo, version v2-revision [software]. doi:10.5281/zenodo.23685897
- [5] ATLAS Collaboration. 2012. Observation of a New Particle in the Search for the Standard Model Higgs Boson with the ATLAS Detector at the LHC. *Phys. Lett. B* (2012). doi:10.1016/j.physletb.2012.08.020
- [6] ATLAS Collaboration. 2014. Dataset from the ATLAS Higgs Boson Machine Learning Challenge 2014. CERN Open Data Portal. doi:10.7483/OPENDATA.ATLAS.ZBP2.M5T8

- [7] ATLAS Collaboration. 2015. Evidence for the Higgs-Boson Yukawa Coupling to Tau Leptons with the ATLAS Detector. *J. High Energy Phys.* 2015, 4 (2015), 117. doi:10.1007/JHEP04(2015)117
- [8] P. Baldi, P. Sadowski, and D. Whiteson. 2014. Searching for Exotic Particles in High-Energy Physics with Deep Learning. *Nat. Commun.* 5 (2014), 4308. doi:10.1038/ncomms5308
- [9] P. Baldi, P. Sadowski, and D. Whiteson. 2015. Enhanced Higgs to $\tau^+\tau^-$ Searches with Deep Learning. *Phys. Rev. Lett.* 114 (2015), 111801. doi:10.1103/PhysRevLett.114.111801
- [10] V. Belis, P. Odagiu, and T. K. Årrestad. 2024. Machine Learning for Anomaly Detection in Particle Physics. *Rev. Phys.* 12 (2024), 100091. doi:10.1016/j.revip.2024.100091
- [11] Y. Burda, R. Grosse, and R. Salakhutdinov. 2016. Importance Weighted Autoencoders. In *International Conference on Learning Representations (ICLR)*. arXiv:1509.00519. <https://arxiv.org/abs/1509.00519>
- [12] O. Cerri, T. Q. Nguyen, M. Pierini, M. Spiropulu, and J.-R. Vlimant. 2019. Variational Autoencoders for New Physics Mining at the Large Hadron Collider. *J. High Energy Phys.* 2019, 5 (2019), 036. doi:10.1007/JHEP05(2019)036
- [13] CMS Collaboration. 2012. Observation of a New Boson at a Mass of 125 GeV with the CMS Experiment at the LHC. *Phys. Lett. B* (2012). doi:10.1016/j.physletb.2012.08.021
- [14] J. H. Collins, K. Howe, and B. Nachman. 2018. Anomaly Detection for Resonant New Physics with Machine Learning. *Phys. Rev. Lett.* 121 (2018), 241803. doi:10.1103/PhysRevLett.121.241803
- [15] B. Efron. 1979. Bootstrap Methods: Another Look at the Jackknife. *Ann. Stat.* 7, 1 (1979). doi:10.1214/aos/1176344552
- [16] M. Farina, Y. Nakai, and D. Shih. 2020. Searching for New Physics with Deep Autoencoders. *Phys. Rev. D* 101 (2020), 075021. doi:10.1103/PhysRevD.101.075021
- [17] T. Finke, M. Krämer, A. Morandini, A. Mück, and I. Oleksiyuk. 2021. Autoencoders for Unsupervised Anomaly Detection in High Energy Physics. *J. High Energy Phys.* 2021, 6 (2021), 161. doi:10.1007/JHEP06(2021)161
- [18] K. Fraser, S. Homiller, R. K. Mishra, B. Ostidek, and M. D. Schwartz. 2022. Challenges for Unsupervised Anomaly Detection in Particle Physics. *J. High Energy Phys.* 2022, 3 (2022), 066. doi:10.1007/JHEP03(2022)066
- [19] E. Govorkova, E. Puljak, T. Aarrestad, T. James, V. Loncar, M. Pierini, et al. 2022. Autoencoders on Field-Programmable Gate Arrays for Real-Time, Unsupervised New Physics Detection at 40 MHz at the Large Hadron Collider. *Nat. Mach. Intell.* 4 (2022), 154–161. doi:10.1038/s42256-022-00441-3
- [20] T. Heimel, G. Kasieczka, T. Plehn, and J. M. Thompson. 2019. QCD or What? *SciPost Phys.* 6, 3 (2019), 030. doi:10.21468/SciPostPhys.6.3.030
- [21] G. E. Hinton and R. R. Salakhutdinov. 2006. Reducing the Dimensionality of Data with Neural Networks. *Science* 313, 5786 (2006), 504–507. doi:10.1126/science.1127647
- [22] G. Kasieczka, B. Nachman, D. Shih, O. Amram, A. Andreassen, et al. 2021. The LHC Olympics 2020: A Community Challenge for Anomaly Detection in High Energy Physics. *Rep. Prog. Phys.* 84, 12 (2021), 124201. doi:10.1088/1361-6633/ac36b9
- [23] G. Kasieczka and D. Shih. 2020. Robust Jet Classifiers through Distance Correlation. *Phys. Rev. Lett.* 125 (2020), 122001. doi:10.1103/PhysRevLett.125.122001
- [24] D. P. Kingma and J. Ba. 2014. Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980. <https://arxiv.org/abs/1412.6980>
- [25] D. P. Kingma and M. Welling. 2013. Auto-Encoding Variational Bayes. arXiv preprint arXiv:1312.6114. <https://arxiv.org/abs/1312.6114>
- [26] F. T. Liu, K. M. Ting, and Z.-H. Zhou. 2008. Isolation Forest. In *2008 Eighth IEEE International Conference on Data Mining (ICDM)*. IEEE, 413–422. doi:10.1109/ICDM.2008.17
- [27] G. Louppe, M. Kagan, and K. Cranmer. 2017. Learning to Pivot with Adversarial Networks. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*. arXiv:1611.01046. <https://arxiv.org/abs/1611.01046>
- [28] L. McInnes, J. Healy, and J. Melville. 2018. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv preprint arXiv:1802.03426. <https://arxiv.org/abs/1802.03426>
- [29] E. M. Metodieiev, B. Nachman, and J. Thaler. 2017. Classification without Labels: Learning from Mixed Samples in High Energy Physics. *J. High Energy Phys.* 2017, 10 (2017), 174. doi:10.1007/JHEP10(2017)174
- [30] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan. 2018. Do Deep Generative Models Know What They Don't Know? arXiv preprint arXiv:1810.09136. <https://arxiv.org/abs/1810.09136>
- [31] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. arXiv preprint arXiv:1912.01703. <https://arxiv.org/abs/1912.01703>
- [32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* 12 (2011), 2825–2830.
- [33] S. Ramaswamy, R. Rastogi, and K. Shim. 2000. Efficient Algorithms for Mining Outliers from Large Data Sets. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. ACM, 427–438. doi:10.1145/342009.335437
- [34] scikit-learn developers. 2026. sklearn.preprocessing.QuantileTransformer. scikit-learn documentation. Accessed 2026-10-13. <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.QuantileTransformer.html>
- [35] J. Shlens. 2014. A Tutorial on Principal Component Analysis. arXiv preprint arXiv:1404.1100. <https://arxiv.org/abs/1404.1100>
- [36] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, et al. 2019. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. arXiv preprint arXiv:1907.10121; published in *Nat. Methods* (2019). doi:10.1038/s41592-019-0686-2

A Additional Checks and Results

The tables and figures in this appendix are referred to in the main text.

A.1 Further checks: details

- (1) *Confirmation sample.* The jet-corrected evaluation is repeated with `sig_clf` in place of `sig_eval`. The detectors never saw `sig_clf`, and it played no part in choosing the correction.
- (2) *Baselines.* The jet count itself used as a score, an isolation forest [26] with 200 trees, and the distance to the 10th nearest of 50,000 training-background events [33].
- (3) *Scores on the latent codes.* The Mahalanobis distance of a code from the mean code of 100,000 training-background events, the negative log-likelihood under an eight-component Gaussian mixture fitted to the same codes, and, for the VAE, the loss of Equation 2 evaluated with the mean code (the negative evidence lower bound).
- (4) *Masked score.* The squared error is averaged only over the features defined for the event, so median-filled jet features and the indicators do not enter the score.
- (5) *No direct jet inputs.* `PRI_jet_num` and the two jet indicators are removed, leaving 30 inputs, and PCA at every k and the AE and VAE at $k = 8$ (three seeds each) are retrained with the same protocol.
- (6) *Correlated energy-scale shifts.* In one half of the test background, all 14 energy-like features (transverse momenta, masses and energy sums), or only the four jet features among them, are multiplied by $f \in \{1.02, 1.05, 1.10, 1.20\}$ before the median fill and the quantile transform. Angles, pseudorapidities, ratios and the jet count are unchanged, so shifted events keep the correlations of real events.

A.2 Further results

The effect of latent size depends on the model and on how many features move. PCA at 18 features gets worse as components are added at every shift of 0.5 standard deviations or more. For example, at a shift of 1 standard deviation the efficiency falls from 7.5% at $k = 2$ to 3.2% at $k = 16$. The AE does best at $k = 8$ for the broad shift (31.0% against 19.8% at $k = 16$). For the narrow shift (3 features, 2 standard deviations) it keeps improving with latent size, from 6.1% at $k = 2$ to 33.8% at $k = 16$, although the spread across seeds and feature sets is large at $k = 16$ (Figure A3a).

The mean R^2 of the feature readout rises with latent size for all models. At $k = 16$ it is 0.892 for PCA, 0.759 for the AE and 0.631 for the VAE. At $k = 8$ the readout varies strongly between

features (Figure A5). The azimuthal-angle features (PRI_tau_phi, PRI_lep_phi, PRI_met_phi) are barely recovered: R^2 is 0.00 to 0.02 for PCA, 0.02 to 0.06 for the AE and 0.18 to 0.22 for the VAE. For the pseudorapidity features PRI_lep_eta and PRI_tau_eta, R^2 is 0.70 and 0.69 for PCA, 0.39 and 0.37 for the VAE, and 0.05 and 0.05 for the AE. The feature PRI_jet_all_pt is recovered well by all three models (0.97 for PCA, 0.95 for the AE and 0.80 for the VAE).

Two-dimensional maps of the codes at $k = 8$ (seed 0, 5,000 background and 5,000 signal events) agree with the probes. In the UMAP maps of the AE and PCA codes, events form separate groups that line up with jet count (Figure A7), and the signal points are spread across those same groups (Figure A6).

Table A2: Event counts in each data split. Number of background and signal events in each split of the ATLAS Higgs challenge dataset [6].

Class	Split	Events
Background	Training	377,074
Background	Validation	80,802
Background	Test	80,802
Signal	sig_clf (reference and probes)	139,780
Signal	sig_eval (evaluation)	139,780

Note. Signal events are never used for training, validation, early stopping, fitting the imputer or transform, or choosing settings; sig_clf is used only by the labeled reference models and sig_eval only for evaluation. Splits made with scikit-learn 1.9.1 (random seed 0).

Table A4: Shifted events caught at a 1% false-positive rate. The autoencoder and variational autoencoder catch synthetic shifts far more often than PCA.

Model	k	3 features	18 features		
		$s = 2$	$s = 0.5$	$s = 1$	$s = 2$
PCA	2	4.4	2.0	7.5	77.5
PCA	4	5.1	1.7	6.0	66.5
PCA	8	7.7	1.6	5.1	57.1
PCA	16	4.6	1.3	3.2	48.7
AE	2	6.1	2.9	16.3	95.7
AE	4	10.3	4.0	18.3	94.8
AE	8	20.6	3.4	31.0	100.0
AE	16	33.8	2.4	19.8	100.0
VAE	2	6.2	2.5	12.5	95.8
VAE	4	10.6	2.5	18.4	97.7
VAE	8	19.2	3.0	22.4	99.6
VAE	16	18.0	2.9	21.0	99.4

Note. Entries are the percentage of shifted test-background events flagged at a 1% false-positive rate (chance is 1%); k is the latent size and s the shift in standard deviations of the scaled features; AE, autoencoder; VAE, variational autoencoder; PCA, principal component analysis. Values are means over seeds (AE and VAE) and, for the 3-feature case, over three random feature sets. Computed with Python 3.11, PyTorch 2.14.1 and scikit-learn 1.9.1.

Table A1: Prior work most relevant to this study. Setting, approach and main finding of each cited study, as stated in its abstract, set against this study.

Study	Setting	Approach	Finding relevant here
Farina, Nakai and Shih [16]	LHC jets; QCD background, top and gluino jets as signals	Deep autoencoder, reconstruction error	Training on background alone improves signal-to-background discrimination
Heimel <i>et al.</i> [20]	Jet substructure	Autoencoder trained on QCD jets, adversarial mass decorrelation	Can identify jets from decays of heavy resonances
Cerri <i>et al.</i> [12]	LHC events	VAE trained on known processes	Flags events from unseen processes as outliers
Finke <i>et al.</i> [17]	Top and QCD jets	Autoencoder, roles of signal and background inverted	Standard setup is not a model-independent tagger
Fraser <i>et al.</i> [18]	Top and W jets against QCD	VAE	Best background representation is not necessarily best for signals
Metodiev, Nachman and Thaler [29]	Quark and gluon jets	Classification without labels	Classifier trained on mixed samples, no labels needed
Collins, Howe and Nachman [14]	Dijet resonance search	Weak supervision (CWoLa)	Turns a modest 2σ excess into a 7σ excess
Kasieczka <i>et al.</i> [22]	LHC Olympics 2020	Community challenge	Black-box datasets for benchmarking model-agnostic methods
Aarrestad <i>et al.</i> [1]	Over 10^9 simulated LHC events	Community challenge	Anomaly scores for model-independent signal regions
Govorkova <i>et al.</i> [19]	Real-time event selection	Autoencoder on FPGA	Detection in as little as 80 ns
Nalisnick <i>et al.</i> [30]	Image datasets	Flows, VAEs, PixelCNNs	Higher likelihood assigned to data from another dataset
This study	ATLAS Higgs challenge, $H \rightarrow \tau\tau$ as proxy	PCA, AE, VAE; jet-count control; probes	Reconstruction error barely detects the proxy signal once jet count is controlled

Note. QCD, quantum chromodynamics (ordinary quark and gluon jets); AE, autoencoder; VAE, variational autoencoder; PCA, principal component analysis; CWoLa, classification without labels; FPGA, field-programmable gate array; $H \rightarrow \tau\tau$, Higgs boson decay to two tau leptons.

Table A3: Further checks of the jet-count finding. Baselines, latent-code scores, a masked score and retraining without jet inputs all give corrected results near chance; only PCA keeps a small edge.

Check	Score or model, k	Raw AUC	Corrected AUC	Corrected eff. (%)
Confirmation (sig_clf)	PCA, 2	0.592	0.524 (0.521–0.526)	1.17 (1.08–1.29)
	PCA, 4	0.614	0.552 (0.549–0.554)	1.59 (1.49–1.71)
	PCA, 8	0.584	0.506 (0.503–0.509)	0.97 (0.89–1.06)
	PCA, 16	0.547	0.457 (0.455–0.459)	0.66 (0.59–0.71)
	AE, VAE (all k)	0.547–0.591	0.471–0.503	0.67–1.21
Baseline	Jet count as the score	0.587	0.500	1.00
	Isolation forest	0.562	0.491	1.13
	Distance to 10th neighbour	0.594	0.491	0.85
Latent-code score	Mahalanobis distance: PCA, 16	0.603	0.549 (0.546–0.552)	1.85 (1.74–1.99)
	Mahalanobis distance: AE, 4	0.568	0.514 ± 0.028	1.33 ± 0.30
	Mahalanobis distance: VAE, 16	0.595	0.552 ± 0.016	1.53 ± 0.80
	Gaussian-mixture NLL: PCA, 4	0.622	0.555 (0.552–0.557)	1.11 (1.02–1.21)
	Gaussian-mixture NLL: AE, 16	0.600	0.514 ± 0.005	1.29 ± 0.13
	Gaussian-mixture NLL: VAE, 4	0.568	0.546 ± 0.019	0.85 ± 0.19
	Negative ELBO: VAE, all k	0.576–0.587	0.487–0.502	0.72–0.88
Masked score	PCA, 4	0.609	0.554	1.60
	AE, VAE (all k)	0.522–0.592	0.473–0.505	0.69–1.27
No jet inputs	PCA, 2	0.589	0.518	1.11
	PCA, 4	0.613	0.546	1.48
	PCA, 8	0.581	0.500	0.84
	PCA, 16	0.542	0.456	0.66
	AE, 8 (3 seeds)	0.567 ± 0.015	0.498 ± 0.013	0.74 ± 0.04
	VAE, 8 (3 seeds)	0.576 ± 0.010	0.509 ± 0.015	0.67 ± 0.11

Note. Evaluated on sig_eval except the confirmation rows, which use sig_clf; 95% bootstrap intervals for single PCA fits are in brackets. Corrected values use the percentile within jet count; eff., signal efficiency at a 1% false-positive rate (chance 1.00%). Latent-code rows show the latent size with the highest corrected AUC for each score and model (negative ELBO: range over all latent sizes); values for every latent size are in results/revision/E_latent_scores.csv in the repository. Ranges are over configuration means; $\pm$ values are standard deviations over seeds. NLL, negative log-likelihood; ELBO, evidence lower bound. Computed with Python 3.11, PyTorch 2.14.1 and scikit-learn 1.9.1.

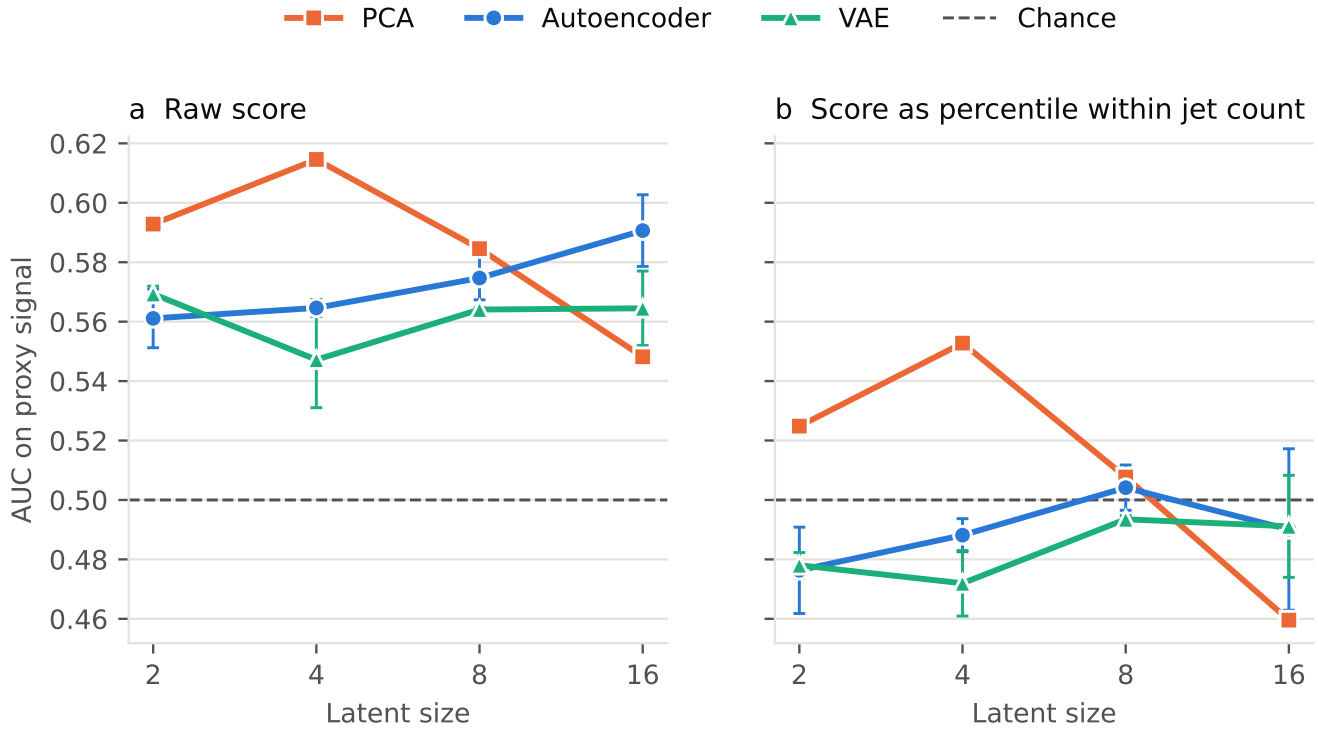


Figure A1: AUC versus latent size, raw and jet-corrected. After the jet-count correction, only PCA stays clearly above chance on the proxy signal.

Note. AUC (area under the receiver operating characteristic curve) against latent size for (a) the raw reconstruction error and (b) the score as a percentile among validation background events with the same jet count; points are means over three seeds with standard-deviation error bars (PCA is a single fit), and the dashed line marks chance. Created with Python 3.11 and matplotlib 3.11.2.

Table A5: Correlated energy-scale shifts. Scaling energies by up to 20% while keeping correlations leaves every detector near chance.

Features scaled	Factor	PCA, 8	AE, 8	VAE, 8	All runs	Max. eff. raw (%)	Max. eff. corrected (%)
All energy-like features	1.02	0.494	0.494	0.492	0.490–0.495	1.03	1.00
	1.05	0.492	0.494	0.491	0.488–0.496	1.02	1.01
	1.10	0.492	0.497	0.495	0.489–0.503	1.05	1.10
	1.20	0.499	0.508	0.512	0.493–0.525	1.22	1.45
Jet features only	1.02	0.497	0.497	0.497	0.495–0.500	1.03	1.09
	1.05	0.496	0.497	0.497	0.494–0.500	1.02	1.10
	1.10	0.495	0.498	0.497	0.492–0.501	1.04	1.14
	1.20	0.496	0.501	0.499	0.491–0.505	1.07	1.19

Note. Entries are raw AUC (area under the receiver operating characteristic curve) for shifted against unshifted halves of the test background; the latent-size-8 columns are means over seeds, and “All runs” gives the range over all 28 runs. The last two columns give the highest share of shifted events caught at a 1% false-positive rate over all runs, raw and after the jet-count correction. Energy-like features are the 14 transverse momenta, masses and energy sums; jet features are the four among them that describe jets. Computed with Python 3.11, PyTorch 2.14.1 and scikit-learn 1.9.1.

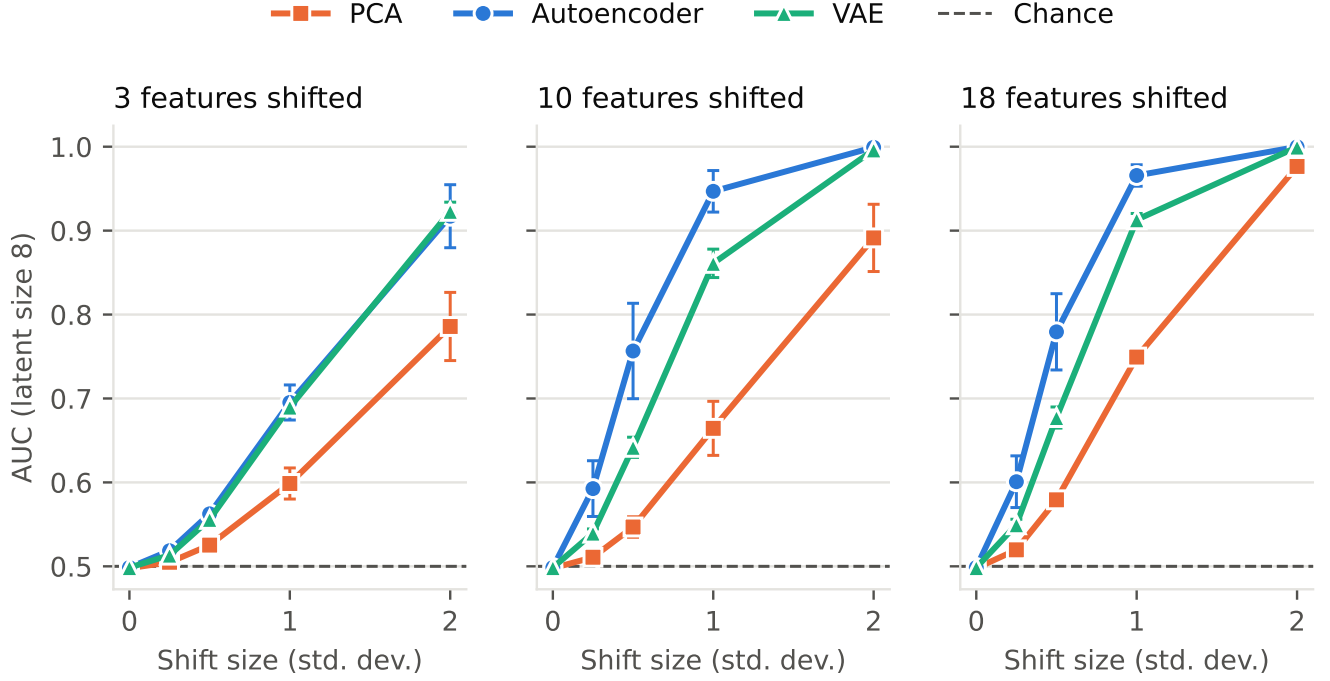


Figure A2: AUC for synthetic shifts at latent size 8. The autoencoder and variational autoencoder detect synthetic shifts at smaller shift sizes than PCA.

Note. AUC (area under the receiver operating characteristic curve) at latent size 8 against shift size in standard deviations, for 3, 10 and 18 shifted features; error bars show the standard deviation over seeds and feature sets, and the dashed line marks chance. Created with Python 3.11 and matplotlib 3.11.2.

Table A6: Linear probes on the latent codes. Linear probes read jet count almost perfectly and the proxy signal partly from the latent codes.

Model	k	Mean R^2	Jet accuracy (%)	Signal AUC	Signal eff. at 1% (%)	Active dims
PCA	2	0.223	93.7	0.590	2.1	–
PCA	4	0.365	94.9	0.656	4.0	–
PCA	8	0.666	99.6	0.759	4.6	–
PCA	16	0.892	99.8	0.762	6.0	–
AE	2	0.170 ± 0.010	97.3 ± 0.8	0.607 ± 0.013	1.1 ± 0.0	–
AE	4	0.214 ± 0.004	99.0 ± 0.9	0.641 ± 0.033	3.1 ± 2.9	–
AE	8	0.392 ± 0.038	100.0 ± 0.0	0.684 ± 0.026	3.0 ± 1.2	–
AE	16	0.759 ± 0.037	100.0 ± 0.0	0.753 ± 0.011	5.9 ± 0.3	–
VAE	2	0.209 ± 0.027	93.8 ± 0.9	0.586 ± 0.030	1.0 ± 0.1	2.0 ± 0.0
VAE	4	0.303 ± 0.062	94.5 ± 3.1	0.687 ± 0.027	2.0 ± 1.0	4.0 ± 0.0
VAE	8	0.549 ± 0.013	100.0 ± 0.0	0.743 ± 0.015	4.3 ± 2.0	7.3 ± 0.6
VAE	16	0.631 ± 0.008	100.0 ± 0.0	0.747 ± 0.018	7.7 ± 2.9	7.0 ± 0.0
Raw inputs	33	–	–	0.801	9.5	–

Note. k , latent size; R^2 , coefficient of determination, averaged over the 18 always-defined features; jet accuracy, share of test events whose jet count is predicted correctly (45.2% for always guessing the most common jet count); signal AUC, area under the receiver operating characteristic curve within jet count; signal eff., signal efficiency at a 1% false-positive rate; active dims, VAE latent dimensions whose mean code has a variance above 0.01. Values are means $\pm$ standard deviation over three seeds (PCA is a single fit); the last row applies the signal probe to the 33 input features. Computed with Python 3.11 and scikit-learn 1.9.1.

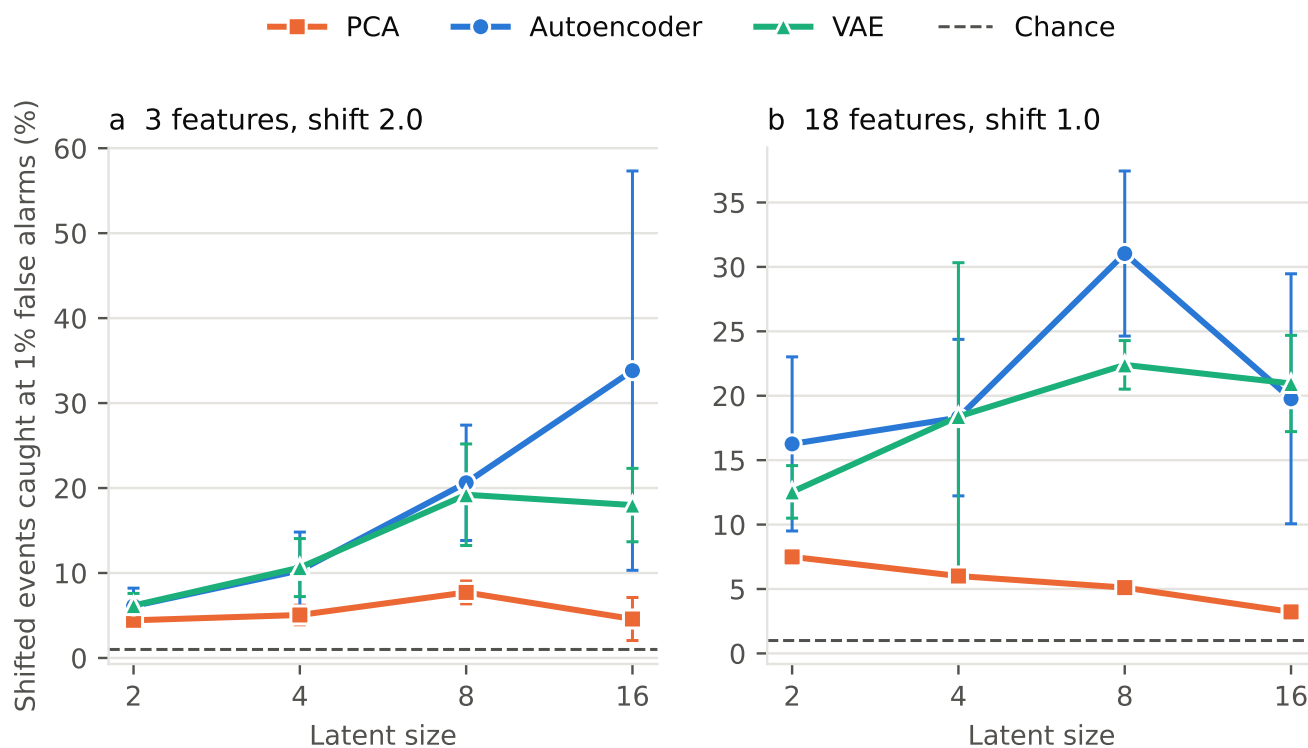


Figure A3: Shift detection by latent size. A larger latent size helps the autoencoder on a narrow shift but makes PCA worse on a broad one.

Note. Share of shifted events caught at a 1% false-positive rate against latent size for (a) three features shifted by 2 standard deviations and (b) all 18 always-defined features shifted by 1 standard deviation; error bars show the standard deviation over seeds and feature sets, and the dashed line marks chance (1%). Created with Python 3.11 and matplotlib 3.11.2.

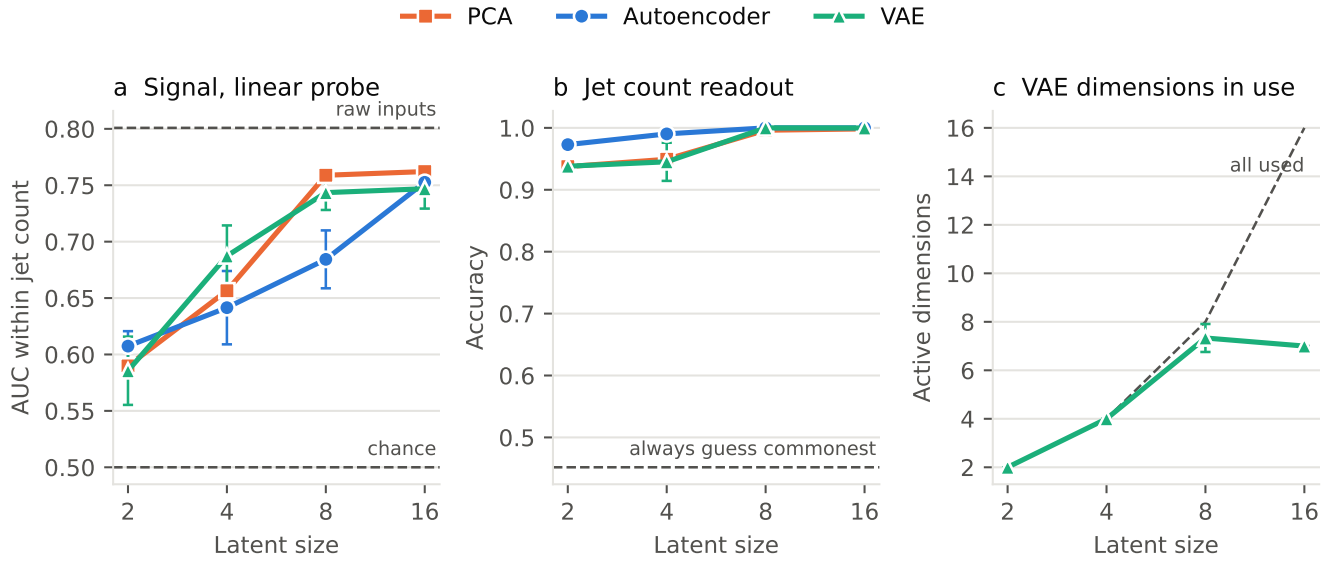


Figure A4: Contents of the latent codes. The latent codes hold the jet count and part of the proxy signal.

Note. (a) AUC (area under the receiver operating characteristic curve) of a linear signal probe within jet count, with dashed lines for chance and for the same probe on the raw inputs; (b) accuracy of reading jet count from the code, with a dashed line for always guessing the most common jet count; (c) active latent dimensions of the variational autoencoder (VAE), with a dashed line where all are used. Points are means over three seeds with standard-deviation error bars. Created with Python 3.11 and matplotlib 3.11.2.

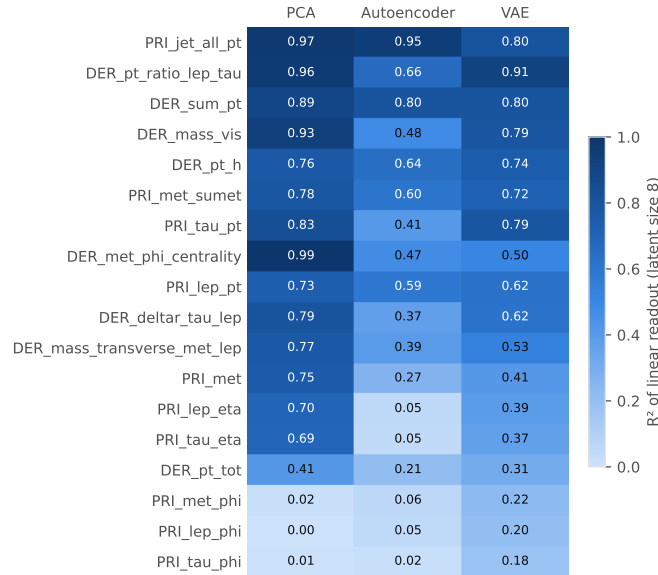


Figure A5: Feature readout from the codes at latent size 8. Azimuthal angles (ϕ) are poorly recovered by every code (R^2 at most 0.22), and the autoencoder also loses the pseudorapidities (η).

Note. Each cell is the R^2 (coefficient of determination) of a linear readout of one always-defined feature from the code at latent size 8, averaged over seeds; features are sorted by their mean R^2 across the three models. Created with Python 3.11 and matplotlib 3.11.2.

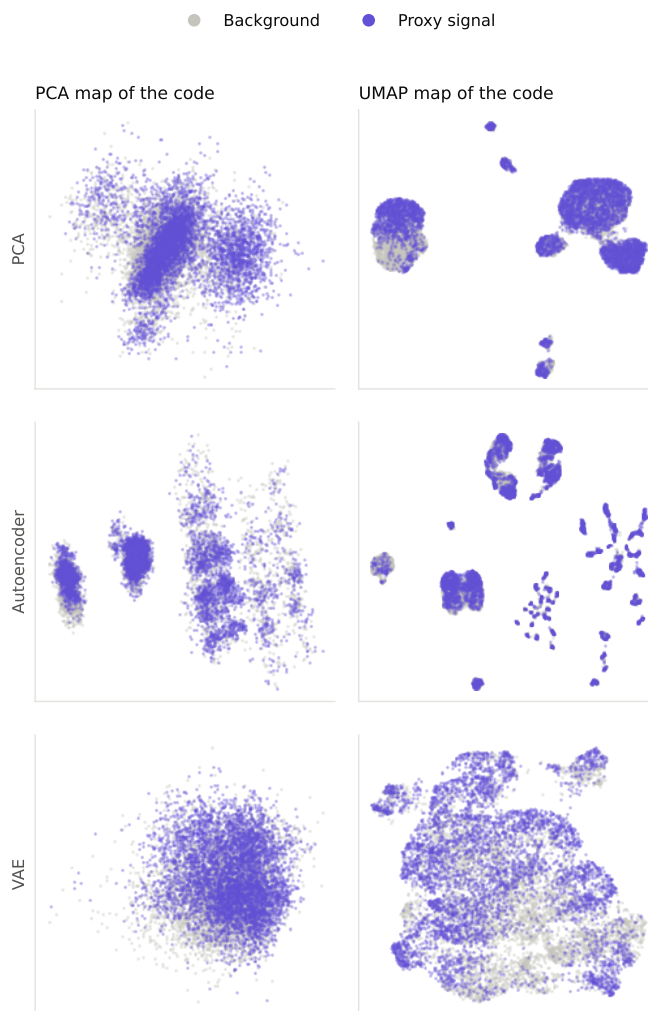


Figure A6: Latent maps colored by class. Proxy-signal points spread across the same groups as background points in every map.

Note. Codes at latent size 8 (seed 0) for 5,000 background and 5,000 signal events, projected to two dimensions with PCA (left) and uniform manifold approximation and projection (UMAP, right); rows are the PCA, autoencoder and variational autoencoder codes, and signal points are drawn over background points. Created with Python 3.11, matplotlib 3.11.2, scikit-learn 1.9.1 and umap-learn 0.5.12.

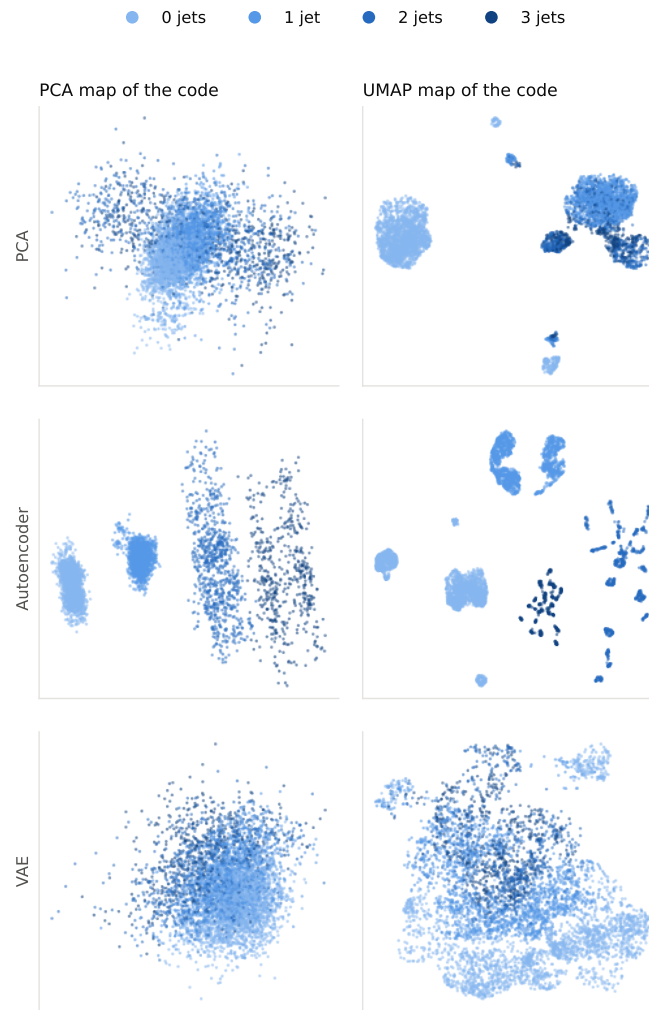


Figure A7: Latent maps colored by jet count. The groups in the latent maps follow jet count.

Note. Background events from the maps in Figure A6, colored by jet count from light (0 jets) to dark (3 jets). Created with Python 3.11, matplotlib 3.11.2, scikit-learn 1.9.1 and umap-learn 0.5.12.