Published for SISSA by Springer

Received: November 16, 2023
Accepted: February 10, 2024
Published: February 28, 2024

Anomaly detection in the presence of irrelevant features

Marat Freytsis,[1] **Maxim Perelstein and Yik Chuen San**

Department of Physics, LEPP, Cornell University,
Ithaca, NY 14853, U.S.A.

E-mail: mf823@cornell.edu, mp325@cornell.edu, ys828@cornell.edu

Abstract: Experiments at particle colliders are the primary source of insight into physics at microscopic scales. Searches at these facilities often rely on optimization of analyses targeting specific models of new physics. Increasingly, however, data-driven model-agnostic approaches based on machine learning are also being explored. A major challenge is that such methods can be highly sensitive to the presence of many irrelevant features in the data. This paper presents Boosted Decision Tree (BDT)-based techniques to improve anomaly detection in the presence of many irrelevant features. First, a BDT classifier is shown to be more robust than neural networks for the Classification Without Labels approach to finding resonant excesses assuming independence of resonant and non-resonant observables. Next, a tree-based probability density estimator using copula transformations demonstrates significant stability and improved performance over normalizing flows as irrelevant features are added. The results make a compelling case for further development of tree-based algorithms for more robust resonant anomaly detection in high energy physics.

Keywords: Automation, Jets and Jet Substructure

ArXiv ePrint: 2310.13057

[1]Present affiliation: Anthropic, San Francisco, California 94960, U.S.A.

https://doi.org/10.1007/JHEP02(2024)220

JHEP02(2024)220

Contents

JHEP02(2024)220

1 Introduction

Experiments at high-energy colliders, such as the Large Hadron Collider (LHC), continue to be the primary source of information about the nature of physics at the microscopic scales. A major task of the current and future experiments is to search for deviations from the Standard Model (SM) of particle physics. Traditionally, such searches are performed by assuming a particular model for physics beyond the Standard Model (BSM), and optimizing the event selection and statistical analysis to obtain maximum sensitivity to the new physics signal in the presence of the SM background. Increasingly, these are supplemented with data-driven methods which minimize model-dependent assumptions about the structure of deviations from the SM, with machine-learning (ML) based approaches the primary driver of such searches [1, 2].

The enormous size and complexity of the data sets collected by collider experiments currently preclude conducting a search for "anything that doesn't look like the SM" in the full data set at once. Even if it were possible in principle, the dependence of collider analyses on complex simulations to interpret measured signals would make such an approach extremely

sensitive to mismodelling errors at all stages of the simulation chain. Instead, recent work focuses on a simpler task of anomaly detection when localized with respect to a particular variable [3–13]. A well-studied benchmark example, starting with the work of [3, 6], is a search for a dijet resonance, in which the signal jets are produced by a boosted resonance decay which is imprinted in non-trivial jet substructure. ML techniques allow for searches of anomalous events with such topology, without making strong model-dependent assumptions about the new physics model that gives rise to this signal. The original algorithm used the Classification Without Labels (CWoLa) approach [14]. In this approach, events are divided into signal and side-band regions based on the invariant mass m_{JJ}. A neural-network (NN) classifier is trained to discriminate between events from the signal and side-band regions. This classifier is then applied to search for anomalous events in the signal region. This approach requires that the features distinguishing signal and background be uncorrelated with m_{JJ}, which is not always the case in real-world applications. To circumvent this problem, algorithms such as ANODE [7] and CATHODE [10] were developed to detect anomalies based on probability density estimation.

A serious issue that can hinder practical applications of the ML-driven anomaly detection methods is the rapid deterioration of performance with growing dimensionality of data space. Typically, collider data contains some observables (or *features*) that are relevant for discriminating signal and background, and a number of observables whose distribution is very similar in the signal and background samples. In a true model-agnostic search, one rarely has the privilege of knowing what features are important beforehand, and inevitably many of the included features can be *irrelevant*. It has been observed that the existing algorithms for anomaly detection, in the context described above, lose their discriminating power very rapidly as even a small number of irrelevant features are added to the input vectors [15]. In this paper, we will present approaches that address this issue within both the classifier-based and probability-density-based approaches to anomaly detection.

The algorithms on which we focus here are based on Boosted Decision Trees (BDTs), rather than neural networks. BDTs tend to outperform neural networks on tabular data, where they can take advantage of the preferred basis implied by the input features [16, 17]. Additionally, given that we are working with meaningful inputs (i.e. high level features), BDTs generally require much less data preprocessing and computational cost compared to neural networks.

The rest of the paper is organized as follows. In section 2, we describe the "signal" and "background" data sets that are used in our analysis, and specify how we model the extraneous irrelevant features. In section 3, we present a BDT-based classifier which uses the CWoLa approach to aid anomaly detection. We show that before irrelevant features are added, the BDT algorithm achieves performance similar to that of NN-based classifiers. However unlike the NN, the BDT performance does not deteriorate significantly when irrelevant features are present. In section 4, we show how the BDT can be used as a probability density estimator, providing a powerful tool for anomaly detection even when relevant features are correlated with m_{JJ}. Furthermore, this algorithm is also robust in the presence of irrelevant features. section 5 contains our conclusions. Technical details related to tuning of hyperparameters of the BDT algorithms are presented in appendices A and B, while a

JHEP02(2024)220

case study of our methods' performance on a dataset with mutually dependent irrelevant features is discussed in appendix C.

In all plots in this paper, the curves showing performance of neural network anomaly-detection tools are generated using code provided at https://github.com/HEPML-AnomalyDetection/CATHODE.

2 Dataset

The signal and background events used in this study are from the LHC Olympics 2020 R&D dataset [18]. In particular, the SM background corresponds to QCD dijet events while the anomalous signal we want to detect is produced by the decay $W' \to X(\to qq)Y(\to qq)$. Here W', X and Y are hypothetical new bosons with masses 3.5 TeV, 500 GeV and 100 GeV respectively. All events are produced using the `Pythia8` [19] and `Delphes 3.4.1` [20] Monte Carlo generators, and jets in each event are identified using `FastJet` [21] using anti-k_T clustering with $R = 1$.

The training (plus validation) set is constructed by combining 1000 randomly selected signal events with a sample of 1 000 000 background events. For evaluation purposes, a separate test set is constructed by having 20 000 signal events and 40 000 background events, all of which lie inside the signal region (defined below). This test set is not used during training.

The physically motivated relevant features are based on the two highest p_T jets. They include

- m_{JJ}: invariant mass of the two jets, which will be the resonant feature.
- m_{J_1}: invariant mass of the lighter jet.
- Δm_J: absolute mass difference between the two jets' invariant masses.
- $\tau_{21}^{J_1}$, $\tau_{21}^{J_2}$: n-subjettiness ratios [22, 23] of the two jets, defined by $\tau_{21} \equiv \tau_2/\tau_1$.

Following [6, 7, 10], we define the signal region (SR) by $m_{JJ} \in [3.3, 3.7]$ TeV, and the sideband region (SB) by $m_{JJ} \notin [3.3, 3.7]$ TeV. Additionally, for the CWoLa method, we also define a *short side-band* (SSB) region, which extends to both sides of the SR by 200 GeV: $m_{JJ} \in ([3.1, 3.3] \cup [3.7, 3.9])$ TeV. These definitions will be used throughout the rest of the paper.

However, for a model-agnostic search, one would not know *a priori* that the observables above are the only features of interest and would likely not have any principled way of excluding additional superfluous features. To simulate such a scenario, we artificially augment the original dataset with features drawn from Gaussian distributions, which will be considered as our irrelevant features. We vary the number of such irrelevant features and examine how much effect they have on anomaly detection performance. Specifically, we study the cases of 4 and 16 irrelevant features. The first case represents the situation where the dataset is a roughly equal mix of relevant and irrelevant features, while the second case provides an example of a dataset dominated by irrelevant features.

JHEP02(2024)220

2.1 What do we mean by irrelevant?

Even though the notion of an irrelevant feature is intuitively clear, it is necessary for us to define it more precisely. We provide here two possible characterizations of ignorable irrelevant features, each suited to the respective anomaly detection method considered in the text.[1] Here "ignorable" means that the feature should not matter in the limit of an infinite amount of data.

- CWoLa method:

  A feature y is irrelevant if $p(m_{JJ} \in \text{SR}|y) = p(m_{JJ} \in \text{SR})$.

- Probability density estimation-based method:

  A feature y is irrelevant if it is statistically independent of m_{JJ} and the auxiliary (relevant) features: $p(m_{JJ}, x_1, \ldots, x_K, y) = p(m_{JJ}, x_1, \ldots, x_K)\, p(y)$. This must hold in both the background and signal samples. Moreover, $p(y)$ in the signal and background samples must be identical.

In this paper, we will explore the performance of anomaly detection algorithms as a function of the number of irrelevant features N. As a baseline model, throughout this paper we assume that the irrelevant features y_i are distributed according to a direct product of Gaussians:

$$p(y_i) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}} e^{-y_i^2/2}. \tag{2.1}$$

A vector of N features drawn from this distribution is then tacked onto each event in the LHCO 2020 dataset described above, with no distinction made between signal and background events. The features y_i in the resulting dataset satisfy both of the irrelevancy definitions above.

Within our baseline model, the y_i's are mutually statistically independent among themselves. This feature is not generic, and is not expected to always hold in realistic physics scenarios. In appendix C we show that our anomaly detection algorithms continue to perform well when the irrelevant features are mutually dependent.

2.2 Performance metric

As is standard, we shall present performance comparisons between NNs and our proposed methods in terms of the *significance improvement characteristic* (SIC) curve, which is obtained by plotting the significance improvement,

$$\text{SIC} = \frac{\epsilon_S}{\sqrt{\epsilon_B}}, \tag{2.2}$$

against ϵ_S. Here ϵ_S is the fraction of correctly identified signal events (true positive rate), and ϵ_B is the fraction of background events incorrectly identified as signals (false positive rate). It should be emphasized here that SIC is a meaningful metric only when the analysis is statistics-limited and not systematics-limited, and the sample is background-dominated; we shall assume that this is the case.

[1] The general definition of irrelevancy has been explored in [24]. The conditions stated below are less general, but suffice in the context of anomaly detection.

JHEP02(2024)220

3 CWoLa on a tree: classifier BDTs

In this section we compare the performance of BDT-based and NN-based CWoLa methods in the presence of irrelevant features.[2]

The CWoLa hunting [3, 6] method attempts to construct the Neyman-Pearson optimal discriminator [25] between a signal and a background where the signal is assumed to be dominantly present in the SR. The key observation underlying this is that if the SSB and SR have different admixtures of signal and background, then the optimal signal-background discriminator is monotonically related to the optimal SSB-SR classifier and finding one produces the other, provided that the auxiliary features $\vec{x}$ are independent of the resonant mass m_{JJ} for the background. While this is a theoretical guarantee, finding an optimal SSB-SR classifier can be difficult in practice. This is because at very low S/B ratio, SSB and SR events largely overlap in feature space with very similar distributions, and most modern machine learning models are flexible enough to mistake local fluctuations for actual excess of signal events (i.e., over-fitting). This situation is particularly exacerbated in the presence of irrelevant features, because they provide additional sources of statistical fluctuations in a higher dimensional space.

The above consideration do not actually select for a method of approximating the SSB-SR classifier. In studies involving the CWoLa hunting method, the classifier typically consists of a fully-connected feedforward neural network. However, it is well-known that neural networks do not fare well with irrelevant inputs, and this is especially so when they are applied in the CWoLa setting for reasons above.

On the other hand, tree-based models are known to be innately robust against irrelevant features [16, 26], an observation usually attributed to the way they are constructed — for most tree-based models they are built by performing cuts in feature space to greedily minimize metrics such as information gain, meaning that they already have some degree of internal feature selection built in. Here we capitalize on this empirical observation and apply BDT-based CWoLa to a more realistic setting where inevitably there will be a lot of irrelevant features.

In what follows, we use `xgboost` as a reference BDT model to compare with a fully-connected feed-forward NN. `xgboost` is chosen since it is widely considered as (one of) the state-of-the-art gradient boosting tree algorithms in terms of speed and accuracy. For detailed descriptions of the `xgboost` algorithm, refer to [27].

3.1 Training procedures

For training the CWoLa classifiers,[3] we select from the raw training set events for which $m_{JJ} \in [3.1, 3.9]$ TeV. This results in roughly 250 000 training events with about 760 signal

[2] In [15], similar comparisons are made in the context of *idealized anomaly detection*, in which perfect understanding of background is assumed. This included a more detailed, physical model of irrelevant features, while we consider a more realistic measurement scenario. We hence view the two studies as naturally complimentary.

[3] The data handling and training procedures are the same as in [10]. Here we summarize them for the sake of completeness.

JHEP02(2024)220

n_estimators	max_depth	eta	alpha	lambda	subsample
292	9	6.2×10^{-3}	50	74	0.75

Table 1. One set of `xgboost` hyperparameters found for the dataset without any irrelevant features. We use default values for other hyperparameters. Refer to appendix A or [27] for a more detailed discussion of these hyperparameters.

events in the SR, which corresponds to $S/B \approx 0.6\%$ and $S/\sqrt{B} \approx 2.2$. Classifiers are then trained to differentiate between SR and SSB labels.

The NN-based classifier is constructed by a fully-connected feed-forward neural network with 3 hidden layers, each of which has 64 neurons. Rectified Linear Unit (ReLU) activation function is used. The network is trained for 100 epochs with binary cross-entropy loss using the `Adam` optimizer [28] and learning rate set to 10^{-3}. During training, only half of the dataset constructed above is used for actual training while the other half is used for validation purposes. In particular, the 10 epochs with the lowest validation error are used to construct an ensemble of 10 classifiers.

For the `xgboost`-based classifier, we employ a 10-fold cross-validation so that the entire training set is utilized during actual training. Specifically, we use the cross-validation process to tune `xgboost`'s hyperparameters to the dataset without irrelevant features.[4] Details of this procedure can be found in appendix A. Since the hyperparameter optimization procedure is stochastic, we find 10 independent sets of hyperparameters, each of which is used to train a separate classifier and they together form an ensemble of 10 classifiers. The same set of hyperparameters is also used to train dataset augmented with irrelevant features. In table 1, we show one set of hyperparameters found.

3.2 Performance comparison

The performances of `xgboost`-based and NN-based CWoLa are shown in figure 1. Both `xgboost`-based and conventional CWoLa perform similarly in the absence of irrelevant features. However, when irrelevant features are present, the performance degradation of the neural network is much more severe than that of the BDT. In particular, in the regime of large number of irrelevant features (relative to number of relevant ones), the neural network-based CWoLa method becomes essentially ineffective. On the other hand, while BDTs-based CWoLa also suffers from the presence of irrelevant features, it is far more resilient. In particular, even with 16 noisy features, the classifier can still attain an average maximum significance improvement of around 7.

In the plots in figure 1, the BDT hyperparameters are optimized once, using the dataset with no irrelevant features, and then kept fixed as the classifier is applied to the datasets with 4 and 16 irrelevant features. This performance can be further improved by dedicated hyperparameter optimization each time more irrelevant features are added. The performances of `xgboost` on the augmented dataset when the hyperparameters are properly tuned are shown in figure 2. Impressively, much of the model's original performance in the absence of irrelevant features can be recovered without too much of computational burden (relative

[4] This is necessary because the default hyperparameters are far too aggressive and lead to severe overfitting.

JHEP02(2024)220

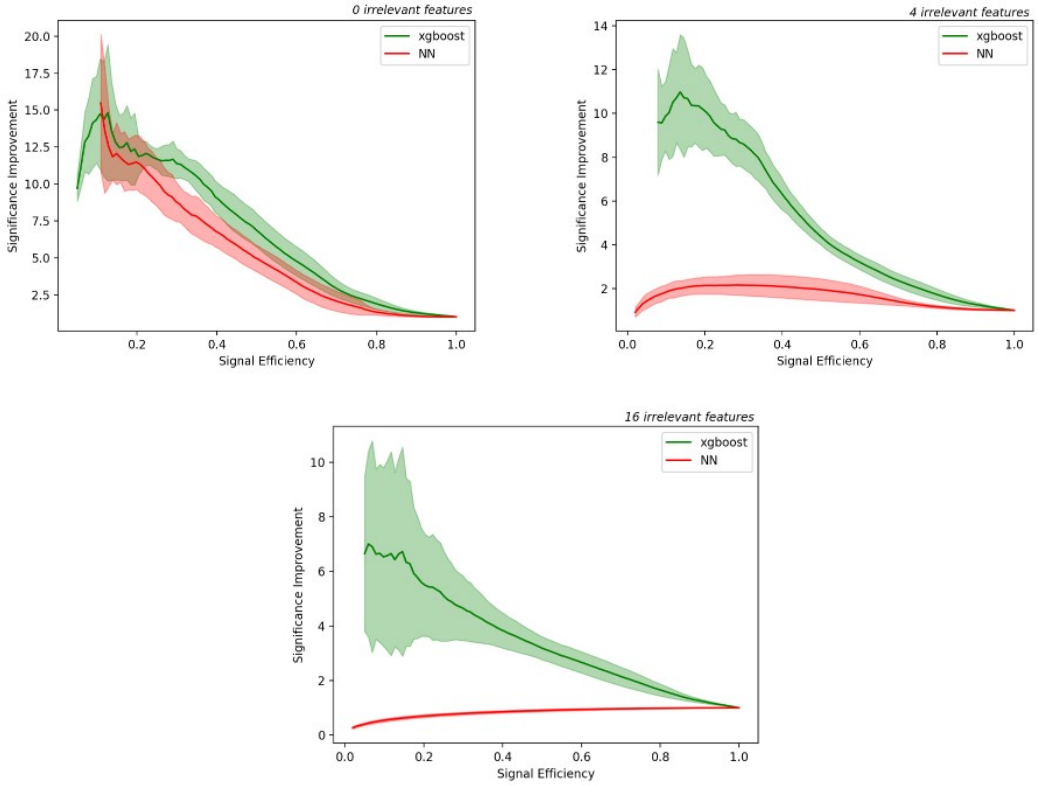

Figure 1. Performance comparisons between BDT-based (green) and NN-based (red) CWoLa methods for dataset augmented with 0, 4, and 16 irrelevant Gaussian features. The same `xgboost` hyperparameters are used to train in all 3 cases. The solid lines represent average SIC value across a classifier ensemble defined in the text, and the bands refer to 1 standard deviation of SIC. It is important to note that for neural networks, the bands correspond to variability for a *fixed* set of hyperparameters, while for `xgboost` they correspond to variability across *different* hyperparameters found by Bayesian optimization. Clearly, `xgboost` is far more robust against the inclusion of irrelevant features than neural networks.

to neural networks). This shows the overall superiority of using BDTs when the input data is of tabular form in the context of CWoLa hunting.

Another added bonus of using a tree-based classifier is that there exists a naturally defined and easily computable notion of *feature importance* [26]. Recall how a tree-based model is constructed: cuts along different feature directions are selected so as to greedily minimize the loss function. Hence, for each feature one can compute how much it contributes to the overall decrease in loss. This adds a layer of interpretability to the model which can potentially be used to shed light on what features are more relevant in discerning signal from background.[5]

We can use this notion of feature importance to understand why `xgboost` is so much more robust compared to neural networks. In figure 3, we show box plots of feature importance values as given by the 10 different classifiers in the ensemble in the case of having 16 irrelevant

[5]It is important to emphasize that this is meaningful only when the features are mostly uncorrelated from each other. If not, it becomes difficult to isolate the effect of each individual feature.

JHEP02(2024)220

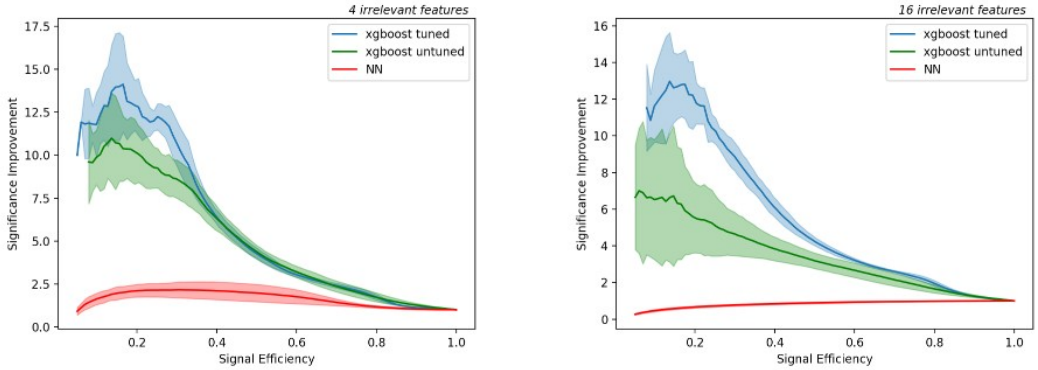

Figure 2. SIC curves for the `xgboost` classifier with hyperparameters optimized for the dataset with no irrelevant features (green), and the same classifier with hyperparameters re-optimized each time more irrelevant features are added (blue). With proper tuning, much of the original performance can be recovered even when the dataset has a large fraction of irrelevant features. NN-based classifier's performance on the same datasets (red) is provided to help guide the eye.

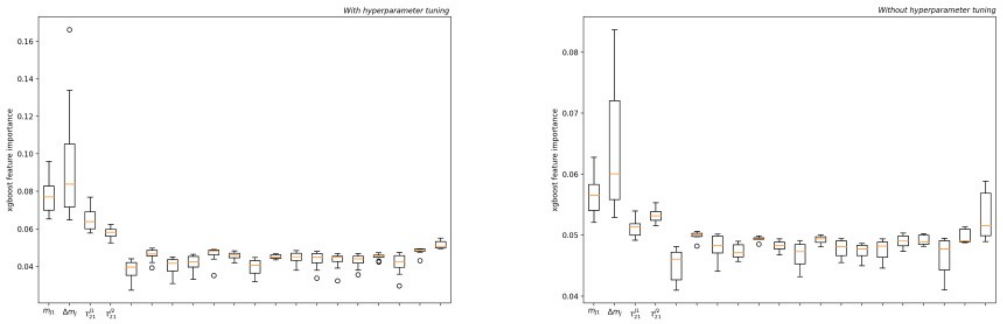

Figure 3. Box plots of feature importance values by 10 independent `xgboost` BDT classifiers applied to dataset with 16 irrelevant features. The BDT hyperparameters are optimized on the same dataset (left panel) or on the dataset with no irrelevant features (right panel). The four labeled boxed on the left side of each plot correspond to relevant features, while the 16 unlabelled boxes correspond to the artificially introduced Gaussian noise features. The relevant features in the original dataset are found by the method to be more important for signal/background discrimination compared to the Gaussian noise, and such a difference is even more pronounced with proper hyperparameter tuning.

features. Strikingly, the model clearly utilizes the relevant features much more than the irrelevant ones, corroborating with our intuition that tree-based models by nature perform a certain degree of internal feature selection. This is likely the reason why the `xgboost`-based CWoLa shows such favorable results.

In conclusion, even a naive direct application of BDT algorithms to CWoLa method can significantly increase its robustness to irrelevant features compared to NN-based CWoLa.

4 Probability density estimation with BDTs

Even though the CWoLa method can achieve significant sensitivity improvement in anomaly detection, its success hinges on the independence of the auxiliary features $\vec{x}$ with m_{JJ} under

JHEP02(2024)220

the background hypothesis, which is quite a strong assumption and does not hold in general. When this assumption is sufficiently violated, CWoLa performance drops drastically [7, 10].

Since this is a strong assumption that does not always hold in physical analyses scenarios of interest, various methods have been proposed to circumvent it. In particular, we examine the anomaly detection with density estimation (ANODE) method [7], which was originally implemented using normalizing flows. While this is not the state-of-the-art anomaly detection method, it is chosen since the lessons learned here can be easily transferred to other similar density-estimation-based methods.

Unlike CWoLa, the ANODE method tries to estimate the two probability densities: $p(\vec{x}|m)$ of the full data set, and $p(\vec{x}|m, \text{bkgd})$ of the background only (estimated from the sideband regions and extrapolated in the signal region). Then, the likelihood ratio

$$R = \frac{p(\vec{x}|m)}{p(\vec{x}|m, \text{bkgd})} \tag{4.1}$$

is computed in the signal region. This ratio can be shown to be optimal (in the Neyman-Pearson sense) without any need of additional assumptions.

In other words, the ANODE method mainly consists of two steps:

- Estimate the full density $p(\vec{x}|m)$ directly from data,
- Estimate the background density $p(\vec{x}|m, \text{bkgd})$ by interpolating from the SB regions into the SR region.

Note that a hidden assumption here is that the auxiliary features have smooth distributions over the SR in the background, for otherwise there would be no reason to believe that interpolation would give a sensible background estimate. This is often true in practice given that the SR is rather small.

Below we explain how the same steps can be achieved using boosted trees.

4.1 Boosted density estimation trees

Motivated by the success of using BDTs with the CWoLa method, here we examine the possibility of applying them to density estimation. Specifically, we follow the tree density estimation algorithm presented in [29], which we describe briefly here. For details, please refer to the original literature.

Conceptually, the BDT density estimation algorithm is very similar to that of normalizing flows [30] — they both model the transformation between the target density and some base density as a composition of simple, bijective maps. Importantly, each composition is thought of as a round of boosting just as in the traditional algorithm.

The major difference between the two is that in the case of BDT, the transformations are built from cuts in the feature space (selected so that they locally minimize the KL divergence between the empirical distribution of the transformed data and the base distribution, which is uniform in our case) with Jacobians admitting closed-form evaluations, whereas for normalizing flows they are typically parameterized by neural networks [30]. The density estimated by the corresponding tree is then a leaf-wise constant function. After each round of boosting, one can define and compute the difference between the learned density and the target density,

JHEP02(2024)220

which is used as the target density for next round. This procedure is recursively performed until some termination condition is satisfied.

4.1.1 Copula

When estimating probability densities, it is often helpful to separate the task of estimating marginal densities and from the task of estimating the dependence structure between variables. This can be achieved explicitly by Sklar's theorem [31], which states, as part of the theorem, that any multivariate probability density $p(x_1, \ldots, x_d)$ (satisfying some very mild conditions) can be represented in the following form:

$$\begin{aligned} p(x_1, \ldots, x_d) &= c(F_1(x_1), \ldots, F_d(x_d)) f_1(x_1) \cdots f_d(x_d) \\ &\equiv \tilde{c}(x_1, \ldots, x_d) f_1(x_1) \cdots f_d(x_d) \,, \end{aligned} \tag{4.2}$$

where the f_i's are the marginal densities, the F_i's are the corresponding cumulative distribution function (CDF), and c is the so-called copula density function. This copula function completely encapsulates the information about dependence structure among variables.

The input data considered in this paper consists of the dijet mass m_{JJ}, the auxiliary features $x_1, \ldots, x_K$, and the additional features $y_1 \ldots y_N$ which contain no information relevant for anomaly detection. Moreover, we assume that irrelevant features are statistically independent of $(m_{JJ}, x_1, \ldots, x_K)$. With this assumption, the copula decomposition takes the form

$$p(m_{JJ}, x_1, \ldots, x_K, y_1, \ldots, y_N) = \tilde{c}(m_{JJ}, x_1, \ldots, x_K)\, \tilde{c}(y_1, \ldots, y_N) \tag{4.3}$$

$$\times\, f_1(x_1) \cdots f_K(x_K) g_1(y_1) \cdots g_N(y_N) \,. \tag{4.4}$$

Furthermore, if the irrelevant features are mutually independent among themselves, as in eq. (2.1), the corresponding copula function is trivial,

$$\tilde{c}(y_1, \ldots, y_N) = 1. \tag{4.5}$$

Then, the likelihood ratio in eq. (4.1) takes the form

$$R = \frac{\tilde{c}(m_{JJ}, x_1, \ldots, x_K)}{\tilde{c}(m_{JJ}, x_1, \ldots, x_K|\text{bkgd})} \prod_{k=1}^{K} \frac{f(x_k)}{f(x_k|\text{bkgd})} \,. \tag{4.6}$$

Note that by using the copula decomposition, the dependence on the irrelevant features in R drops out in both the marginal and the copula densities. The cancellation in the marginal density ratio is easy to ensure in practice since it simply relies on univariate density estimation. As for the copula density, the model needs to be able to learn that it is independent of $(y_1 \ldots y_N)$. This is where the tree-structure shines — similar to the supervised case, the tree model should be able to learn to not cut along the irrelevant directions, since they do not contribute much towards the decrease in KL divergence when estimating the copula density.

In view of the discussion above, we follow the basic two-stage strategy suggested in [29]: we first fit models to the marginal variables, and then we use the learned CDF to transform them to the copula space on which we estimate the corresponding copula density. The final learned density is given by eq. (4.2). Note that neither the copula factorization, eq. (4.3),

JHEP02(2024)220

	n_estimators	max_depth	lr	gamma
marginal	100	10	0.1	0.3
copula	2500	50	0.1	0.3

Table 2. Hyperparameters used to estimate the marginal and copula densities with the tree-based algorithm in [29]. Please refer to the original literature or appendix B for meanings of these parameters.

nor the mutual independence of the irrelevant features, eq. (4.5), are hardwired into our algorithm. Rather, these features are efficiently learned by the BDT from the structure of the training data. The high quality of the trained tree model contributes to robustness of the anomaly detection algorithm in the presence of irrelevant features. At the same time, the underlying BDT has sufficient flexibility to remain useful when the structure of the input data is more complex. This is evidenced by the example with mutually dependent irrelevant features considered in appendix C.

4.2 Interpolation

Once the probability density in the SB region is estimated, the next step is to interpolate it into the SR. Unlike the NN, a tree-based density estimator does not automatically provide such an interpolation, and it needs to be implemented by hand. This represents an additional step in the algorithm, but has an inherent advantage of being controllable, in contrast to a black box-like interpolation performed by the NN. As a baseline, we employ a naive linear interpolation:

$$p(\vec{x}|m) = p(\vec{x}|m_L) + \frac{p(\vec{x}|m_R) - p(\vec{x}|m_L)}{m_R - m_L}(m - m_L), \quad m \in (m_L, m_R)\,, \tag{4.7}$$

where $m \equiv m_{JJ}$; m_L and m_R are the lower and upper boundaries of the signal region in m_{JJ}; and the vector $\vec{x}$ includes both auxiliary (relevant) and irrelevant features. While more elaborate methods of interpolation exist [8, 11, 32], this simple form is chosen here for the following reasons:

- Under the assumption that the SR is sufficiently small and that the SB is not significantly signal-contaminated, we expect linear interpolation to give reasonable results (there are however some subtleties, see section 4.5).
- More importantly, in eq. (4.7), the interpolated density is explicitly linear in the learned density. In the ideal case that the irrelevant features' densities factorize from the learned density, this property ensures that dependence on irrelevant variables will be cancelled out in the construction of the likelihood ratio. As we shall see below, this linearity property is important in ensuring robustness.

4.3 Training and evaluation procedures

The NN-based density estimator we use in our comparison is a masked-autoregressive flow (MAF). The training procedure for a MAF is the same as in [10], and we refer readers to the original paper for details.

The training of the tree-based density estimator is done by feeding the algorithm the entire training dataset (560 000 events), with hyperparameters listed in table 2. The performance of

JHEP02(2024)220

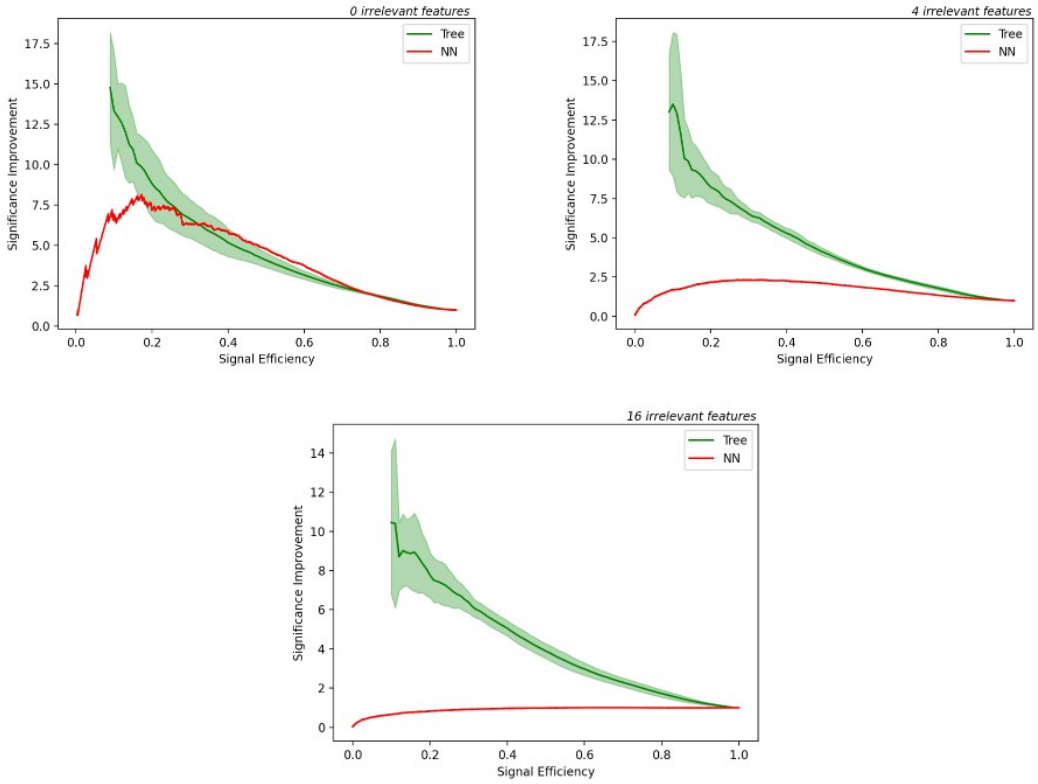

Figure 4. Performance comparison between the BDT implementation of density estimator (green) and the NN-based implementation found in [29] (red), for the original LHCO dataset. The error bands show 1 standard deviation of significance improvement across 10 random training-validation-test splits. The BDT implementation shows superior performance both with and without inclusion of irrelevant features. In particular, even with 16 irrelevant features added, the BDT only shows a small level of degradation.

the BDT density estimation algorithm is fairly insensitive to the choice of hyperparameters as long as the resulting model is sufficiently expressive, due to the fact that we are estimating the density from the background sample for which we have large statistics. (This is in contrast with the CWoLa case, where the BDT needs to learn a small difference between the distributions of auxiliary features in the signal and side-band regions.) The preprocessing of [10] is not necessary in this case, since trees are invariant under monotonic transformations. The trained model is then used to evaluate $p(\vec{x}|m)$, $p(\vec{x}|m_L)$, and $p(\vec{x}|m_R)$. The latter two are used to estimate the background density in the signal region according to eq. (4.7).

The performance of each method is evaluated on a separate test set consisting of 20 000 signal events and 60 000 background events, all of which lie in the SR. In particular, we train the tree-based model for 10 random training-validation-test splits to cross-validate its variance. The same is not done for neural networks due to their high computational costs. Comparisons are shown in the next section.

4.4 Performance comparison

In figure 4, we show the performance comparisons between MAF- and tree-based density estimation algorithms. Without irrelevant features, the tree-based algorithm already provides

JHEP02(2024)220

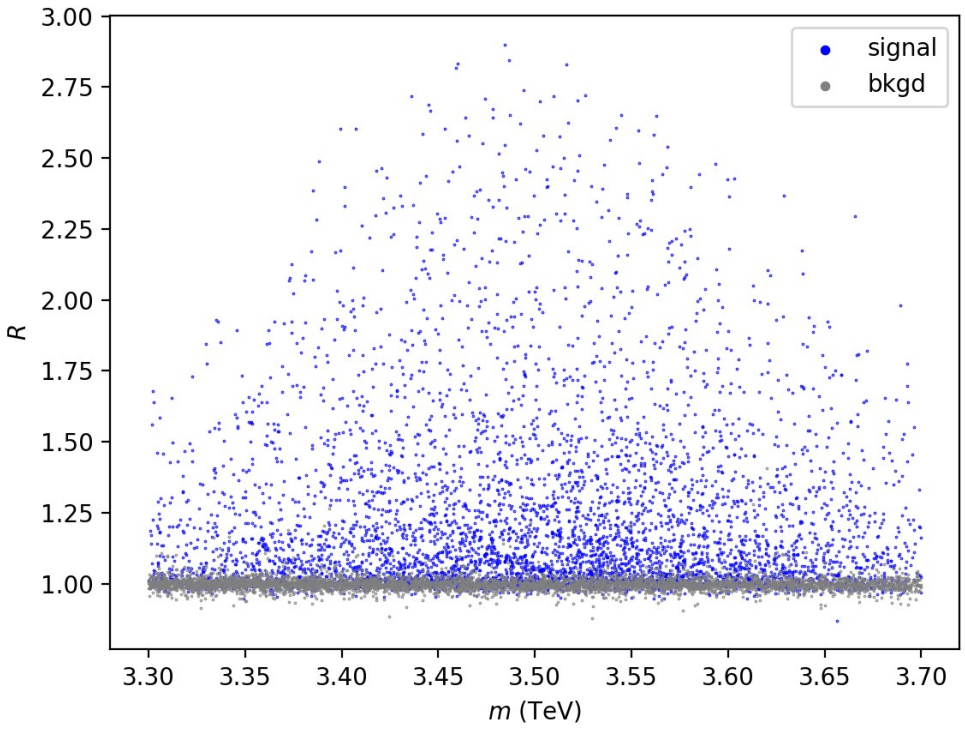

Figure 5. Scatter plot of the density ratio R defined in eq. (4.1) against m_{jj} for 5000 signal events and 5000 background events in the test set (signal region). The plot shows that a naive linear interpolation is sufficient for the LHCO dataset.

a significant improvement over the NN in the low signal efficiency region. Furthermore, just as in the case of CWoLa, the MAF-based algorithm suffers from severe performance degradation as the number of irrelevant features increases. In the case where 16 irrelevant features are added, the method is essentially no different from a random classifier. On the other hand, the tree-based algorithm is remarkably robust, showing almost no degradation of performance with up to 16 irrelevant features.

As an additional note, the success of the tree-based density estimation algorithm also shows that a simple linear interpolation for background estimation is very effective, at least for the LHCO dataset. In figure 5 we show a scatter plot of the data-to-background density ratio against m_{jj} for both signal and background events. It can be clearly seen that the simple linear interpolation is effective in estimating the background density for the LCHO dataset. We believe this is evidence that more considerations should go into studying the interpolation method instead of relying on a black box like NNs.

4.5 Correleated auxiliary features

The primary motivation for density estimation methods is to address situations where the CWoLa assumption of statistical independence between the auxiliary features $\vec{x}$ and m_{JJ} does not hold. However, in the example considered above, $\vec{x}$ and m_{JJ} were independent to a large degree. In this section we explore how our strategies proposed above perform when $\vec{x}$ and m_{JJ}

JHEP02(2024)220

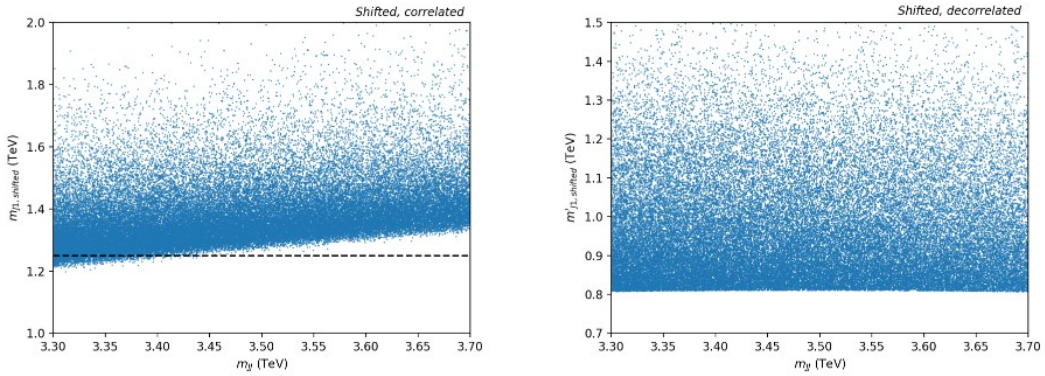

Figure 6. Scatter plot of m_{J1} against m_{JJ} for the log-shifted dataset over the signal region. Before decorrelation (left panel), the plot clearly shows why a naive linear interpolation should fail — the interpolation over the dashed line crosses the support of data density, which would cause a sharp change in the interpolated density. After decorrelation (right panel), we see that the support is roughly axis-parallel, and we expect that a simple linear interpolation should suffice.

are not independent. Specifically, we artificially introduce dependence between $\vec{x}$ and m_{JJ} via[6]

$$m_{J_1} \to m_{J_1} + \log m_{JJ}, \tag{4.8}$$

$$\Delta m_J \to \Delta m_J + \log m_{JJ}, \tag{4.9}$$

where all the masses are measured in units of TeV.

In this case, we immediately see a difficulty with our proposed interpolation method. When $\vec{x}$ and m_{JJ} are strongly dependent, the support of $p(m, \vec{x})$ can be of arbitrary shape in general, but the interpolation in eq. (4.7) implicitly assumes that for a fixed $\vec{x}$, $p(\vec{x}|m)$ does not vary too much as a function of m_{JJ} across the SR.[7] This is illustrated in the left panel of figure 6, where a naive linear interpolation over the dashed line would result in an abrupt and unphysical drop in the interpolated density. This situation can be handled automatically by NNs since they are able to perform more global interpolations, but we need to be more careful when implementing the interpolation by hand.

It is clear from the above discussion that the quality of linear interpolation eq. (4.7) requires that $\vec{x}$ and m_{JJ} be roughly independent *over the SR*. To achieve this, we perform the following simple "decorrelation" procedure. For each feature x, consider the following transformation:[8]

$$m \to m\,, \tag{4.10}$$

$$x \to f(x, m)\,, \tag{4.11}$$

[6]We consider a non-polynomial dependence on m_{JJ} instead of a linear one, as in [7, 10], because our decorrelation scheme below will be able to completely undo linear correlation, thus making the comparison not very useful.

[7]Note that this is different from the CWoLa assumption since we only require weak dependence over the SR. In general this is easier to attain.

[8]We do not transform m_{JJ} since this is a privileged variable under the localized-signal assumption.

JHEP02(2024)220

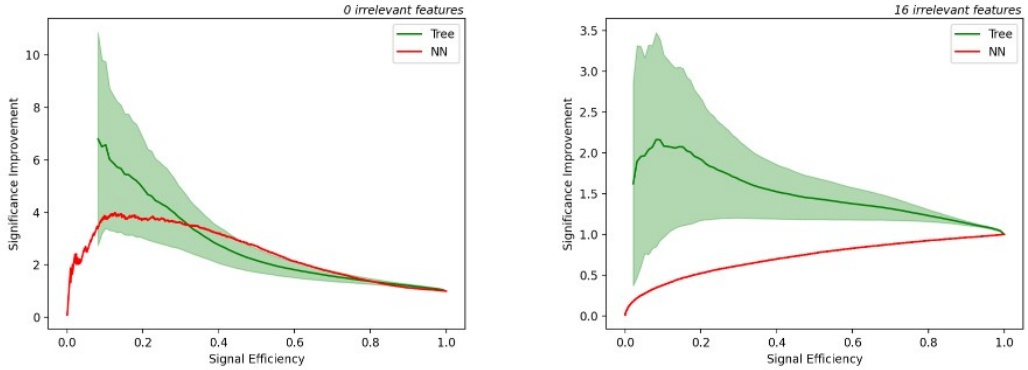

Figure 7. The SIC curves for NN-based (red) and tree-based (green) density-estimation algorithms applied to the log-shifted dataset. The cases with 0 (left panel) and 16 (right panel) irrelevant features are shown for comparison. The error bands are defined in the same way as in figure 4.

where f is such that the transformation is bijective so that no information carried in x is lost. We can then search for f such that the dependence between x and m_{JJ} within the SR, as measured by distance correlation, is minimized. In particular, we consider f belonging to a family of functions of the form

$$f(x, m) = a_0(m) + a_1(m)x. \tag{4.12}$$

Since m_{JJ} lies within the SR which we assume to be small, we further parameterize the coefficients a_0 and a_1 as[9]

$$a_0(m) = \alpha m + \beta m^2, \quad a_1(m) = 1 + \gamma m + \delta m^2\,. \tag{4.13}$$

To summarize, we search for values of α, β, γ and δ that minimize the corresponding distance correlation. This minimization is performed using the L-BFGS method [33] implemented in `SciPy` [34].

The right panel of figure 6 shows the scatter plot of m_{J1} against m_{JJ} over the SR after our decorrelation procedure. Visually we can observe that the support of data density is now parallel to the m_{JJ}-axis, and numerically we can achieve a distance correlation of order 10^{-4} between auxiliary features and m_{JJ} over the SR. This signals the success of our decorrelation scheme.

With decorrelation carried out, the rest of the algorithm remains the same as in the previous section. In figure 7 we compare the SIC curves of the NN-based and tree-based algorithms applied to the log-shifted dataset. We observe that the tree-level algorithm still greatly outperforms the NN-based method when irrelevant features are added. At the same time, we also note that the performance of the tree-based algorithm is not as robust with respect to addition of irrelevant features as in the unshifted case (see section 4.4). This is likely due to the decorrelation procedure above, which by chance will find non-zero $(\alpha, \beta, \gamma, \delta)$ such that the in-sample distance correlation between m_{JJ} and the transformed irrelevant feature

[9]Without loss of generality, we can take the constant term in a_0 to be zero and the constant term in a_1 to be 1, since distance correlation remains invariant under such a choice.

JHEP02(2024)220

is minimized. This effect can in principle be mitigated by more rigorous cross-validation technique, but we do not pursue this point here.

While the simple approach to decorrelation and interpolation taken in this paper is effective, it may be seen as somewhat *ad hoc*. Many more elaborate interpolation methods exist in the literature (e.g., Gaussian process regression, high dimensional splines [26]), which may further improve the performance and robustness of our algorithm. We leave the exploration of such methods for future work.

5 Discussion and conclusions

In this work, we have presented two tree-based approaches to detect anomalies in the presence of irrelevant features. Anomaly detection methods are already starting to be used in LHC analyses, with searches based on CWoLa hunting at ATLAS already released [35]. Since BDT-based methods are already used in experimental analyses, we hope that our methods would be readily able to be adopted and calibrated for experimental use. We first considered a CWoLa-inspired method, and showed that boosted decision trees are more robust to irrelevant features compared to neural networks. By exploiting the inherent feature selection of decision trees, the BDT-based classifier maintained good performance even with the addition of significantly more irrelevant than discriminating auxiliary features.

In analogy to density estimation methods like ANODE, we proposed using tree-based models paired with a copula transformation and interpolation step. By estimating the marginal and copula densities separately, irrelevant features can be factorized out of the likelihood ratio assuming their mutual independence. Even when this is not the case, we observe that the resulting reduction in significance improvement still leaves the tree-based approach much less sensitive to the presence of these features. Our results demonstrated the promising performance of the tree-based density estimator compared to normalizing flows, especially in higher dimensionality with many irrelevant features. The tree-based model allows for a simple and effective linear interpolation scheme for estimating the background density.

Recently, [15] also explored the use of BDTs for anomaly detection in high-energy collider analyses. This study includes a larger and more physical set of irrelevant features, while also finding increasingly improved performance and greater stability as irrelevant features are added during training. However, it assumes that a perfect sample of the background is available and does not deal with the extrapolation of such a model into a resonant region, as we do. We thus view our results as complementary and together making a compelling case for the application of tree-based methods to anomaly detection.

Overall, tree-based methods seem well-suited for anomaly detection tasks when operating on high-level observables with potential irrelevant features. These naturally lend themselves to presentation as tabular data. The techniques presented here could find useful application in collider searches and other physics analyses aiming to be robust against the embedding of low-dimensional signals in high-dimensional feature spaces. More advanced interpolation schemes than what we consider here might improve the performance and stability of the density-based approach, while exploring other tree-based algorithms like Bayesian Additive Regression Trees might improve the overall fidelity of the learned functions. We leave these possibilities to future work.

JHEP02(2024)220

Acknowledgments

We would like to thank Ben Nachman and David Shih for useful discussions. This research is supported by the NSF grant PHY-2014071. YCS is partially supported by the Boochever Fellowship at Cornell University.

A Hyperparameter tuning for `xgboost`

Here we provide details regarding hyperparameter tuning for the `xgboost` model used in CWoLa method. The hyperparameters we choose to optimize are as follows:[10]

- `n_estimators`: this controls the number of boosting rounds
- `max_depth`: this controls how complex the base tree learner is by limiting how deep each tree can be
- `eta`: this controls how much each tree contributes in building the ensemble
- `alpha`: L_1 regularizer on weights of the model
- `lambda`: L_2 regularizer on weights of the model

`xgboost` has a lot of other parameters, but here we choose to focus on these few because (i) `n_estimators`, `max_depth` and `eta` are known to have the most impact on the model's performance, and (ii) `alpha` and `lambda` explicitly control the model's weights, and therefore they have a direct impact on how much the model will overfit, which is exactly our concern here. In addition, we (arbitrarily) fix the `subsample` parameter, which measures how much of the training data is used in fitting each individual tree, to be 0.75. In principle one can also include it in the hyperparameter search but our empirical results show that the final performance is not very dependent on its exact value. We use default values for all other hyperparameters.

To search for the optimal hyperparameters, we perform Bayesian optimization on the 10-fold cross-validation score, which we define to be the true positive rate of SR-SB labels at a fixed false positive rate of 10^{-3}. Specifically, the Bayesian optimization is carried out using the `gp_minimize` function in the `scikit-optimize` library, with default settings except we reduce the number of calls to 30 in order to save time.

B Hyperparameters of boosted density estimation tree algorithm

Here we describe briefly meanings of some of the hyperparameters used when training on the LHCO dataset. Please refer to [29] for details.

- `n_estimators`: number of boosting rounds
- `max_depth`: maximum depth each base tree learner can grow to

[10] More details about these hyperparameters can be found in [27].

JHEP02(2024)220

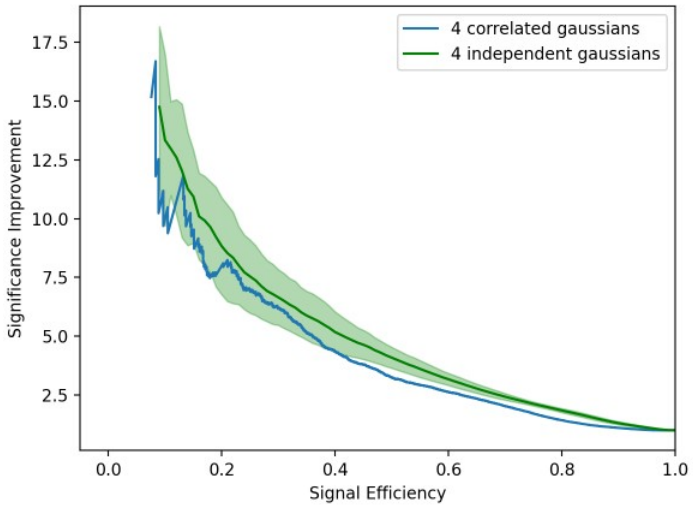

Figure 8. SIC curve of the tree-based density estimation algorithm with mutually correlated irrelevant features (blue), compared to the baseline case of mutually independent irrelevant features studied in section 4 (green).

- `lr`: global shrinkage parameter that helps smooth out density learned during each boosting round. When it is equal to 0, each tree returns the uniform base distribution (no learning); when it is equal to 1, each tree returns the empirical distribution (most aggressive learning).

- `gamma`: amount of node-specific shrinkage. When it is 0, only the global learning rate `lr` is used; when it is a positive real number, the amount of shrinkage for each node grows as its volume in feature space decreases.

C Mutually dependent irrelevant features

In the baseline model used throughout this paper, the irrelevant features enjoy the extra property that they are mutually independent, see eq. (2.1). While this extra property has no bearing on the CWoLa hunting method, it does affect our use of copula in section 4: if the irrelevant features $\vec{y}$ are mutually independent, the copula density c becomes independent of $\vec{y}$. While such independence was not hardwired into our algorithm, it can potentially make the copula density easier to learn, and one might wonder how robust the algorithm is if irrelevant features are mutually dependent.

To test this, we rotate the original irrelevant features by a random matrix A:

$$\vec{y}\,' = A\vec{y}\,. \tag{C.1}$$

Here A is constructed by independently sampling each of its elements from the standard normal distribution. The elements of the rotated irrelevant feature vector $\vec{y}\,'$ are now mutually dependent. We then apply the tree-based density estimation algorithm described in section 4 to the dataset $(m_{JJ}, \vec{x}, \vec{y}\,')$, where $\vec{x}$ are the relevant auxiliary features.

JHEP02(2024)220

The resulting SIC curve is shown in figure 8. We can see that our method's performance is very similar to the case considered in the main text, demonstrating that the method's performance is not reliant on the factorization property of eq. (2.1). In other words, the BDT is able to learn the non-trivial copula function involving irrelevant features well enough to not cause any degradation in the overall performance. In future work, it would be interesting to further test this aspect of the algorithm in realistic physical applications of anomaly detection.

JHEP02(2024)220

References

[1] G. Kasieczka et al., *The LHC Olympics 2020 a community challenge for anomaly detection in high energy physics*, *Rept. Prog. Phys.* **84** (2021) 124201 [arXiv:2101.08320] [INSPIRE].

[2] T. Aarrestad et al., *The Dark Machines Anomaly Score Challenge: Benchmark Data and Model Independent Event Classification for the Large Hadron Collider*, *SciPost Phys.* **12** (2022) 043 [arXiv:2105.14027] [INSPIRE].

[3] J.H. Collins, K. Howe and B. Nachman, *Anomaly Detection for Resonant New Physics with Machine Learning*, *Phys. Rev. Lett.* **121** (2018) 241803 [arXiv:1805.02664] [INSPIRE].

[4] T. Heimel, G. Kasieczka, T. Plehn and J.M. Thompson, *QCD or What?*, *SciPost Phys.* **6** (2019) 030 [arXiv:1808.08979] [INSPIRE].

[5] M. Farina, Y. Nakai and D. Shih, *Searching for New Physics with Deep Autoencoders*, *Phys. Rev. D* **101** (2020) 075021 [arXiv:1808.08992] [INSPIRE].

[6] J.H. Collins, K. Howe and B. Nachman, *Extending the search for new resonances with machine learning*, *Phys. Rev. D* **99** (2019) 014038 [arXiv:1902.02634] [INSPIRE].

[7] B. Nachman and D. Shih, *Anomaly Detection with Density Estimation*, *Phys. Rev. D* **101** (2020) 075042 [arXiv:2001.04990] [INSPIRE].

[8] A. Andreassen, B. Nachman and D. Shih, *Simulation Assisted Likelihood-free Anomaly Detection*, *Phys. Rev. D* **101** (2020) 095004 [arXiv:2001.05001] [INSPIRE].

[9] K. Benkendorfer, L.L. Pottier and B. Nachman, *Simulation-assisted decorrelation for resonant anomaly detection*, *Phys. Rev. D* **104** (2021) 035003 [arXiv:2009.02205] [INSPIRE].

[10] A. Hallin et al., *Classifying anomalies through outer density estimation*, *Phys. Rev. D* **106** (2022) 055006 [arXiv:2109.00546] [INSPIRE].

[11] J.A. Raine, S. Klein, D. Sengupta and T. Golling, *CURTAINs for your sliding window: Constructing unobserved regions by transforming adjacent intervals*, *Front. Big Data* **6** (2023) 899345 [arXiv:2203.09470] [INSPIRE].

[12] A. Hallin et al., *Resonant anomaly detection without background sculpting*, *Phys. Rev. D* **107** (2023) 114012 [arXiv:2210.14924] [INSPIRE].

[13] T. Golling, S. Klein, R. Mastandrea and B. Nachman, *Flow-enhanced transportation for anomaly detection*, *Phys. Rev. D* **107** (2023) 096025 [arXiv:2212.11285] [INSPIRE].

[14] E.M. Metodiev, B. Nachman and J. Thaler, *Classification without labels: Learning from mixed samples in high energy physics*, *JHEP* **10** (2017) 174 [arXiv:1708.02949] [INSPIRE].

[15] T. Finke et al., *Back To The Roots: Tree-Based Algorithms for Weakly Supervised Anomaly Detection*, arXiv:2309.13111 [INSPIRE].

[16] L. Grinsztajn, E. Oyallon and G. Varoquaux, *Why do tree-based models still outperform deep learning on typical tabular data?*, in *Advances in Neural Information Processing Systems 35: 36th Conference on Neural Information Processing Systems (NeurIPS 2022)*, S. Koyejo et al. eds., Curran Associates Inc. (2022), pp. 507–520 [https://proceedings.neurips.cc/paper_files/paper/2022/file/0378c7692da36807bdec87ab043cdadc-Paper-Datasets_and_Benchmarks.pdf].

[17] V. Borisov et al., *Deep Neural Networks and Tabular Data: A Survey*, arXiv:2110.01889 [DOI:10.1109/TNNLS.2022.3229161].

[18] G. Kasieczka, B. Nachman and D. Shih, *R&D Dataset for LHC Olympics 2020 Anomaly Detection Challenge*, (2019) [DOI:10.5281/zenodo.6466204].

[19] C. Bierlich et al., *A comprehensive guide to the physics and usage of PYTHIA 8.3*, *SciPost Phys. Codeb.* **2022** (2022) 8 [arXiv:2203.11601] [INSPIRE].

[20] DELPHES 3 collaboration, *DELPHES 3, A modular framework for fast simulation of a generic collider experiment*, *JHEP* **02** (2014) 057 [arXiv:1307.6346] [INSPIRE].

[21] M. Cacciari, G.P. Salam and G. Soyez, *FastJet User Manual*, *Eur. Phys. J. C* **72** (2012) 1896 [arXiv:1111.6097] [INSPIRE].

[22] J. Thaler and K. Van Tilburg, *Identifying Boosted Objects with N-subjettiness*, *JHEP* **03** (2011) 015 [arXiv:1011.2268] [INSPIRE].

[23] J. Thaler and K. Van Tilburg, *Maximizing Boosted Top Identification by Minimizing N-subjettiness*, *JHEP* **02** (2012) 093 [arXiv:1108.2701] [INSPIRE].

[24] G.H. John, R. Kohavi and K. Pfleger, *Irrelevant Features and the Subset Selection Problem*, in *Machine Learning Proceedings 1994*, W.W. Cohen and H. Hirsh Elsevier (1994), p. 121–129 [DOI:10.1016/b978-1-55860-335-6.50023-4].

[25] J. Neyman and E.S. Pearson, *On the Problem of the Most Efficient Tests of Statistical Hypotheses*, *Phil. Trans. Roy. Soc. Lond. A* **231** (1933) 289 [INSPIRE].

[26] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning*, Springer (2009) [DOI:10.1007/978-0-387-84858-7] [INSPIRE].

[27] T. Chen and C. Guestrin, *XGBoost: A Scalable Tree Boosting System*, arXiv:1603.02754 [DOI:10.1145/2939672.2939785] [INSPIRE].

[28] D.P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, arXiv:1412.6980 [INSPIRE].

[29] N. Awaya and L. Ma, *Unsupervised tree boosting for learning probability distributions*, arXiv:2101.11083.

[30] G. Papamakarios et al., *Normalizing Flows for Probabilistic Modeling and Inference*, arXiv:1912.02762 [INSPIRE].

[31] A. Sklar, *Fonctions de répartition à n dimensions et leurs marges*, *Publ. Inst. Stat. Univ. Paris* **8** (1959) 229.

[32] D. Sengupta, S. Klein, J.A. Raine and T. Golling, *CURTAINs Flows For Flows: Constructing Unobserved Regions with Maximum Likelihood Estimation*, arXiv:2305.04646 [INSPIRE].

[33] D.C. Liu and J. Nocedal, *On the limited memory BFGS method for large scale optimization*, *Math. Programming* **45** (1989) 503 [INSPIRE].

JHEP02(2024)220

[34] P. Virtanen et al., *SciPy 1.0–Fundamental Algorithms for Scientific Computing in Python*, *Nature Meth.* **17** (2020) 261 [arXiv:1907.10121] [INSPIRE].

[35] ATLAS collaboration, *Dijet resonance search with weak supervision using $\sqrt{s} = 13$ TeV pp collisions in the ATLAS detector*, *Phys. Rev. Lett.* **125** (2020) 131801 [arXiv:2005.02983] [INSPIRE].

JHEP02(2024)220