Feature selection with distance correlation

Ranit Das,[1,*] Gregor Kasieczka,[2,3,†] and David Shih[1,‡]

[1]*NHETC, Department of Physics and Astronomy, Rutgers University, Piscataway, New Jersey 08854, USA*
[2]*Institut für Experimentalphysik, Universität Hamburg, 22761 Hamburg, Germany*
[3]*Center for Data and Computing in Natural Sciences (CDCS), 22607 Hamburg, Germany*

(Received 10 March 2023; accepted 12 January 2024; published 6 March 2024)

Choosing which properties of the data to use as input to multivariate decision algorithms—also known as feature selection—is an important step in solving any problem with machine learning. While there is a clear trend towards training sophisticated deep networks on large numbers of relatively unprocessed inputs (so-called automated feature engineering), for many tasks in physics, sets of theoretically well-motivated and well-understood features already exist. Working with such features can bring many benefits, including greater interpretability, reduced training and run time, and enhanced stability and robustness. We develop a new feature selection method based on distance correlation, and demonstrate its effectiveness on the tasks of boosted top- and W-tagging. Using our method to select features from a set of over 7,000 energy flow polynomials, we show that we can match the performance of much deeper architectures, by using only ten features and two orders-of-magnitude fewer model parameters.

DOI: 10.1103/PhysRevD.109.054009

I. INTRODUCTION

Recently there has been enormous progress in training supervised deep learning classifiers to perform object and event identification at the LHC. Deep learning classifiers that make use of low-level information (such as the four vectors of all the reconstructed particles in a jet or event) have been shown to achieve impressive performance gains over cut-based methods and shallow classifiers trained on high level kinematic features, translating directly into better physics performance [1–3].

One very fruitful benchmark task for developing new architectures has been boosted top tagging, i.e., classifying jets from hadronic-decays of boosted top quarks against the background of light quark and gluon jets. Boosted top jets have a rich, varied and subtle substructure that deep learning classifiers can leverage and exploit to enhance their performance. Boosted top tagging has been a fertile canvas for working with a wide variety of deep learning methods, such as deep neural networks (DNNs) [4–6], convolutional neural networks (CNNs) [7–9], recurrent [10] and recursive neural networks (NNs) [11,12], sets [13], graph NNs [14–16], and transformers [17,18]. Performance gains have also been reported using approaches that exploit the underlying Lorentz invariance [19–23].

However, all of these high-performing deep learning methods are black boxes, and there has been a parallel effort in AI interpretability/explainability to understand "what the machine learns" [24–28]. Recently, an important step in this direction came from [29], which developed a new forward feature selection technique to efficiently scan through more than 7,000 energy flow polynomials (EFPs) [30]—i.e., quantities that measure the energy distribution inside a jet—in order to identify a small number (typically of order ten) that together reproduce as closely as possible the performance of a state-of-the-art black-box NN classifier. Their method relied on a score called "average decision ordering" (ADO) which measures how often a given feature has the same decision ordering (DO) as the reference classifier. This method has been applied to W-jets [29], muons [31], electrons [32], and semivisible dark jets [33].

Aside from shedding light on "what the machine learns", constructive feature selection methods can have several other interesting applications. Classifiers based on high-level features (HLFs) could be more robust against domain shifts and more easy to calibrate with collider data (as a smaller number of distributions need to be validated). Also, a classifier trained on only a few inputs could be made much more lightweight (far fewer parameters), leading to less intensive training and faster evaluation time. This could have important applications to machine learning with microsecond inference times, e.g., for the LHC trigger. Finally, even if attempting to replicate a state-of-the-art deep

[*] ranit@physics.rutgers.edu
[†] gregor.kasieczka@uni-hamburg.de
[‡] shih@physics.rutgers.edu

learning classifier with a set of HLFs falls short, it might have important physics implications, as it could teach us that the set of HLFs being used is incomplete and does not fully capture all the correlations in the data.

In this paper, inspired by [29], we present a new method for forward feature selection. It is based on the measure of statistical independence called distance correlation (DisCo) [34–37], which was first used in the high energy physics (HEP) literature to decorrelate top taggers against jet mass [38], and was subsequently applied to ABCD background estimation [39] and anomaly detection [40]. We use DisCo (instead of ADO) to measure how relevant (statistically dependent) a given set of features is for the classifier output. We show that our DisCo-based forward feature selection method outperforms [29] on both hadronic W tagging and hadronic-top tagging, in the sense that it selects features more efficiently, ultimately achieving better performance with fewer features. The upshot is that on top tagging, our method selects as few as nine EFPs (from the same sample of $7,000+$ as [29]), and training a very compact DNN on these small number of EFPs, we achieve nearly state-of-the-art performance, matching the rejection power of ParticleNet-Lite [14] with only a fraction of parameters.

Importantly, our method does not require a previously obtained reference classifier, but can also be trained equally well *ab initio*, using the truth labels (0 for background and 1 for signal). This is unlike the method of [29], whose performance suffered when trained on truth labels. Therefore, our DisCo-based forward feature selection method is able to operate in two, conceptually different modes: (1) either as an *ab initio* feature selector that aims to produce the best-possible classifier given a set of features; or (2) as a feature selector that aims to "explain" a previously obtained black box classifier.

Note that the proposed forward or constructive feature selection is very different from backward elimination methods which try to iteratively remove features starting with the full set of features, or feature attribution methods which use Shapley values [41–51] to assign contributions of each feature to explain the outcome of a pretrained classifier output. As we will see in the numerical examples, the performance of a classifier trained on the full space of $\approx 7,000$ features is much lower than what a carefully selected set of ≈ 10 features can achieve, further motivating the forward-feature selection strategy.

In the following, we first introduce a strategy for forward feature selection in Sec. II and show how DisCo can be used as a scoring function for promising features. Section III next discusses the concrete application to top tagging. We show that our method reaches performance equal to much more complex architectures, using only a fraction of features and complexity, even matching LorentzNet [22] in ablation studies. There, we also investigate the leading eight EFPs chosen (as well as their stability under repeated application of our method) and attempt to use them to understand what the machine learns. We observe that the same leading six EFPs are found under multiple iterations of our method, indicating their relevance for this task. Finally, Sec. IV provides a discussion of results and further outlook.

II. METHOD

For supervised classification tasks,[1] forward feature selection methods operate on a feature space

$$\mathcal{F} = \{f_1, f_2, f_3, \ldots, f_N\}. \tag{1}$$

We should think of each feature f_i as a predetermined function (e.g., an EFP) that operates on the low-level data $\vec{x} \in \mathbb{R}^d$ of each event, i.e., $f_i = f_i(\vec{x})$. Given an already selected set of n features $\mathcal{F}_n = \{f_{i_1}, f_{i_2}, \ldots, f_{i_n}\}$, the goal of forward feature selection is to identify the next feature $f_{i_{n+1}}$ which is expected to improve the performance on the classification task the most.

It is assumed here that the full feature space $\mathcal{F}$ is so large, and the training of the classifier sufficiently expensive, that one cannot just brute force select the next feature by training $N - n$ classifiers on all possible additional features $f_i \notin \mathcal{F}_n$. Therefore, what is needed here is a much cheaper-to-compute relevance score, that stands in as a proxy for the classifier itself.

The relevance score takes as input a given set of features, together with a reference label, evaluated over the dataset. The reference label could be either truth labels, in which case we are performing *ab initio* forward feature selection in order to produce the highest-performing classifier that we can or the reference label could be a pretrained state-of-the-art classifier, in which case we are performing forward feature selection for the purposes of AI explainability (explaining the pretrained black box classifier).

In any event, for a set of features, the point is that the relevance score can be obtained much more quickly than training a classifier on the features, and the forward feature selection algorithm can select the feature with the highest score as the next feature.

The four steps involved in our feature selection algorithm are illustrated in Fig. 1 and explained in the following:

(1) *Step 1: Train on known features*

Train a classifier network on a set of features $\mathcal{F}_n = \{f_{i_1}, f_{i_2}, \ldots f_{i_n}\}$ using the full training sample of all events X_{all}, and obtain the classifier output y_{pred} for all events in X_{all}.

For simplicity and best possible performance, we use a dense neural network (details in Appendix B), although any other classification algorithm (e.g., XGBoost, logistic regressor) could be used as well.

[1] In this work, we focus on binary classification as the most widely studied task, but generalization of the proposed technique to other supervised learning problems is straightforward.

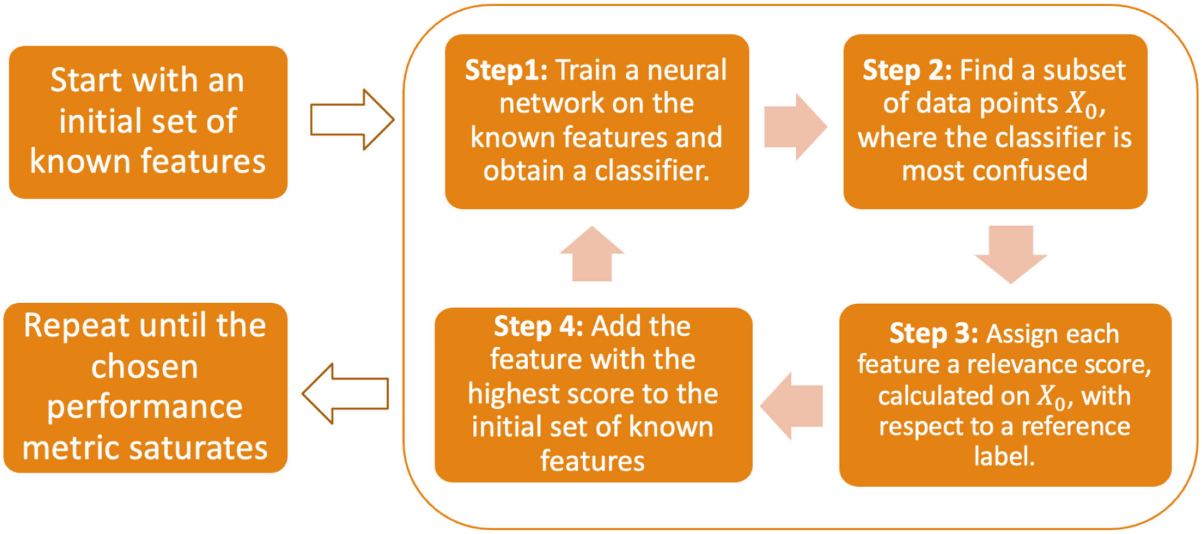

Start with an initial set of known features
Step1: Train a neural network on the known features and obtain a classifier.
Step 2: Find a subset of data points X_0, where the classifier is most confused
Step 3: Assign each feature a relevance score, calculated on X_0, with respect to a reference label.
Step 4: Add the feature with the highest score to the initial set of known features
Repeat until the chosen performance metric saturates

FIG. 1. Overview of the proposed forward-feature selection algorithm.

(2) *Step 2: Select the confusion set $X_0 \subset X_{all}$*

Instead of calculating the relevance scores using the full dataset, we choose to instead focus on a subset of the full data $X_0 \subset X_{all}$ that we call the confusion set. These are events where we believe the features in $\mathcal{F}_n$ are least effective in separating signal from background, and where adding a new feature may have the largest impact. To identify this subset, we select all events in a window around $y_{\rm pred} = 0.5$, as shown in Fig. 2; these should be the events where the classifier is most confused about whether it is a signal or a background. We observe that using a confusion set instead of the full dataset improves performance. We use events in $0.3 < y_{\rm pred} < 0.7$ as our confusion set X_0. The boundaries of this window are important hyperparameters of our algorithm, and we settled on this choice after scanning through different window sizes and seeing where the performance of the method was best.

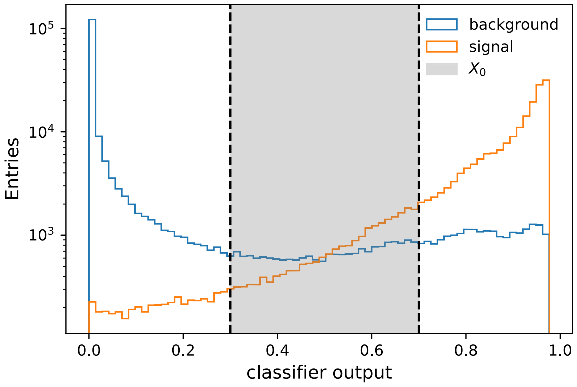


FIG. 2. Events in a window around the classifier output value $y_{\rm pred} = 0.5$ are selected as the confusion set X_0 for DisCo-FFS.

(3) *Step 3: Assign a relevance score to each feature*

To each feature f_i in the feature space $\mathcal{F}$, we assign a relevance score s_{f_i}, which gauges how much the feature will improve classification performance.

The relevance score is calculated using the feature vectors evaluated on the events in the confusion set X_0, together with a reference label $y_{\rm ref}$,

$$\mathcal{X} = \{(f_{i_1}(\vec{x}), \ldots, f_{i_n}(\vec{x}), f_i(\vec{x}))|\vec{x} \in X_0\},$$
$$\mathcal{Y} = \{y_{\rm ref}(\vec{x})|\vec{x} \in X_0\}. \tag{2}$$

The relevance score assigned to each feature f_i is

$$s_{f_i} = \text{Affine-DisCo}(\mathcal{X}, \mathcal{Y}). \tag{3}$$

As described in the Introduction, DisCo is short for distance correlation [34–37], a measure of statistical dependence that is zero if and only if the random vectors $\mathcal{X}$ and $\mathcal{Y}$ are statistically independent, and positive (and ≤ 1) otherwise. Therefore, it is well-suited to judging whether adding f_i to the feature vector $(f_{i_1}, \ldots f_{i_n})$ produces a stronger correlation with the reference label $y_{\rm ref}$ or not. Here we are using the affine-invariant version of DisCo [52], which is invariant under arbitrary linear transformations of $\mathcal{X}$ and $\mathcal{Y}$, in order to make it more robust against basis reparametrizations in the EFP space. The multivariate affine-DisCo calculation is described in more detail in Appendix C.

(4) *Step 4: Add the feature with best relevance score to the list of known features*

We select the feature with the best score and add it to $\mathcal{F}_n$. Then we proceed back to the first step to train

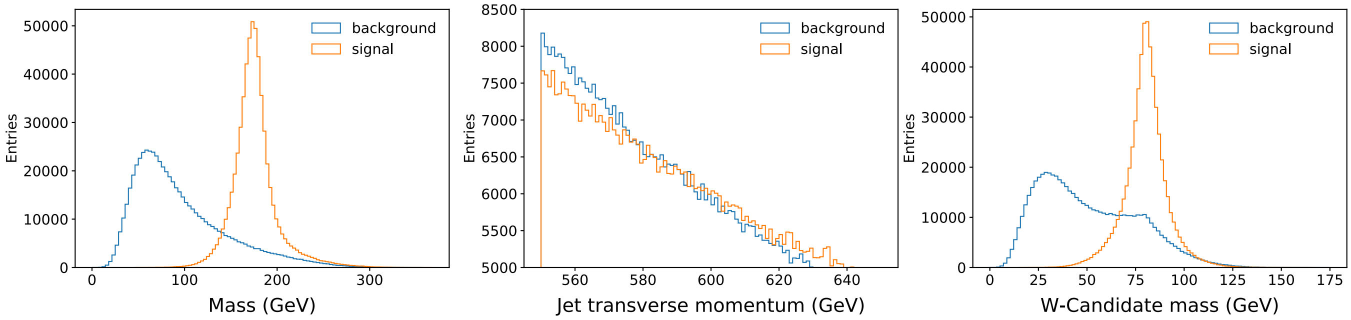


FIG. 3. Initial features chosen for top tagging; jet mass m_J (left), jet p_T (center), and mass of the W-candidate (right).

a network on the updated set of features $\mathcal{F}_{n+1}$. The procedure is stopped when the performance metric saturates and the final set of features is returned.

While the above method explicitly describes our DisCo-based forward feature selection (DisCo-FFS) algorithm, the protocol is general enough to accommodate also other iterative feature selection techniques. In Appendix A, we use the same framework to outline how the forward feature selection from [29] operates. This is based on DO for the confusion set, and ADO for the relevance score, and we will refer to it as DO-ADO-FFS throughout this work.

III. APPLICATION TO TOP-TAGGING

A. Dataset

We study the performance of the DisCo-feature selection algorithms on the top quark tagging landscape dataset [1,53]. This dataset contains boosted, hadronically decaying top jets as signal, and QCD (i.e., light quark and gluon) jets as background, which are generated using PYTHIA8 [54], with a center-of-mass energy of 14 TeV. Multiple interactions and pileup are not included in this dataset. The detector simulation is done using DELPHES [55], with the ATLAS detector card. FastJet [56] is used to create jets using the anti-k_T algorithm [57] with $R = 0.8$. Only jets in the p_T range [500, 650] GeV, and $|\eta_j| < 2$, are considered. The dataset contains only kinematic information, in the form of energy-momentum four-vectors of all the reconstructed particles in each jet, which are extracted using the DELPHES energy-flow algorithm. No additional tracking information or particle information is included.

The full dataset contains 2 million events, with 1 million signal events and 1 million background events. This data is split into 1.2 million events in the training set, 400,000 in the validation set, and 400,000 in the test set, each set containing equal number of signal and background events.

B. Feature space

For top-tagging we start with

$$\mathcal{F}_{\text{initial}} = \mathcal{F}_3 = \{m_J, p_T, m_{W-\text{candidate}}\}, \qquad (4)$$

where m_J is the mass of the jet, p_T is the transverse momentum of the jet and $m_{W-\text{candidate}}$ is the mass of the W-candidate in the jet, calculated with a very simple method; we recluster each fat jet using the exclusive k_T algorithm with $R = 0.3$ into exactly three subjets. Then we pick the pair of subjets whose invariant mass comes closest to m_W. This pair of subjets gives us the W-candidate and their mass is $m_{W-\text{candidate}}$. The distributions of the initial features are illustrated in Fig. 3.

We then apply feature selection algorithms to a large set of EFPs [30]. EFPs are functions of energy fractions and angular separation of jet constituents,

$$z_a^{(\kappa)} = \left(\frac{p_{Ta}}{\sum_{i\in J} p_{Ti}}\right)^{\kappa}, \qquad \theta_{ab}^{(\beta)} = (\Delta\eta_{ab}^2 + \Delta\phi_{ab}^2)^{\beta/2}, \quad (5)$$

where p_{Ta} is the transverse momentum of the ath jet constituent, and the denominator in z_a is summed over all jet constituents in a jet J. EFPs have a one-to-one correspondence with a graph G,

$$\sum_{a\in J} z_a^{(\kappa)} \to (\text{each node}), \qquad \theta_{ab}^{(\beta)} \to (\text{each edge}). \quad (6)$$

Thus, given a graph G, with N nodes and edges $(m,\ell) \in G$, the EFP is

$$\text{EFP}_G^{(\kappa,\beta)} = \sum_{i_1\in J} \cdots \sum_{i_N\in J} z_{i_1}^{(\kappa)} \cdots z_{i_N}^{(\kappa)} \prod_{(m,\ell)\in G} \theta_{i_m i_\ell}^{(\beta)}. \quad (7)$$

The original EFPs [30] were introduced as IRC-safe observables, with $\kappa = 1$. However, in our feature space we are motivated by [29] to consider other values of κ as well. Following [29],[2] we use energy flow polynomials

[2] With one exception, we do not include additional features from $d = 8$ with $c = 4$, as [29] do in their analysis. These features were initially omitted due to difficulties in their calculation. It was later verified that their inclusion does not significantly alter the performance of DisCo-FFS.

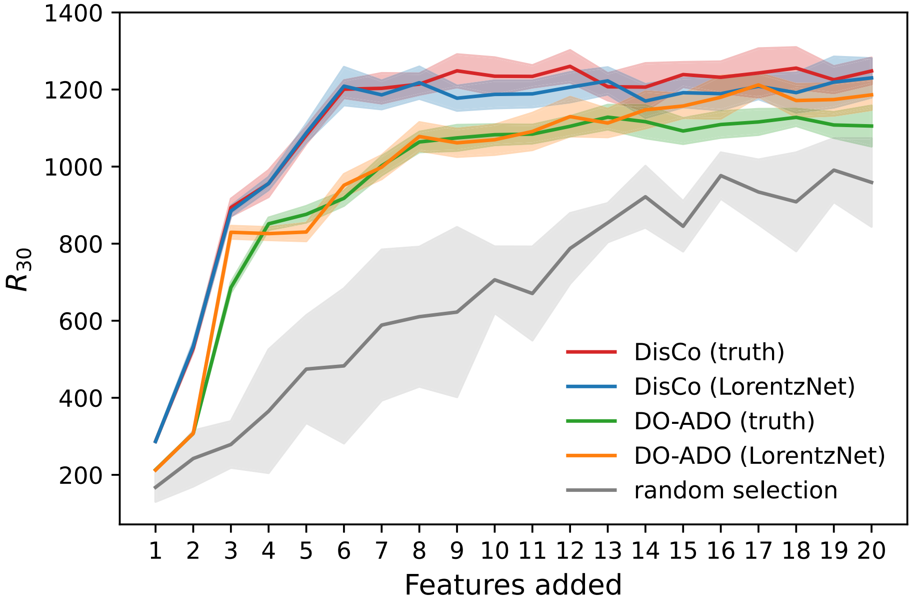


FIG. 4. Performance comparison between DisCo-FFS and DO-ADO-FFS methods, truth-guided and `LorentzNet`-guided. Shown in gray is also the random selection baseline. The shaded bands around each curve come from training the NN classifier ten times on the same set of features (similar to [1]). Overall, DisCo-FFS seems to select more relevant features than DO-ADO-FFS, resulting in a higher-performing classifier at every step. Interestingly, while DO-ADO-FFS with truth labels actually performs *worse* than with `LorentzNet` (a phenomenon also observed in [29]), no degradation in performance is observed for DisCo-FFS with truth labels.

with all combinations of $d \leq 7$, $\beta = [0.5, 1, 2]$ and $\kappa = [-1, 0, 0.5, 1, 2]$, which form a space of 7,320 unique features.

C. Results

1. Ab initio feature selection using truth labels

First, we consider the *ab initio* feature selection task, using the truth labels to guide the algorithms so as to yield the best possible classifier.

We apply the truth-guided DisCo-FFS and DO-ADO-FFS[3] to the training and validation set, and use the test set only for evaluating the performance. (Network architectures and hyperparameters used in this section are described in Appendix B.) The performance metric chosen for top-tagging is R_{30} (the QCD rejection factor at 30% top-tagging efficiency). It allows a better separation of different methods as area under curve (AUC) saturates and is more indicative of the performance at a potential working point.

As shown in Fig. 4, the R_{30} value increases as more features are added using the two feature selection methods. This shows that both DisCo-FFS and DO-ADO-FFS are selecting useful features. After nine features the performance of the features added using the DisCo method saturates with $R_{30} \approx 1250$. We also see that our proposed method outperforms DO-ADO-FFS and achieves a higher R_{30} at each step.

Any worthwhile feature selection algorithm should do better than randomly selecting features. To test this, we randomly select each number of features 10 times, and use the average and standard deviation of the R_{30} as our "random baseline" shown in Fig. 4. Interestingly, we see that the randomly selecting EFPs can also give better performance, as we add more and more features, but not as good as the FFS methods.

2. Feature selection using pretrained classifier

Next we turn to feature selection using a pretrained classifier (so-called black-box guiding in [29]). For the pretrained classifier, we use the state-of-the-art `LorentzNet` tagger [22].

We see in Fig. 4 that DO-ADO-FFS with `LorentzNet` actually performs slightly better than DO-ADO-FFS with truth labels. This somewhat counterintuitive result was also observed by [29] in the context of boosted W-tagging, and we confirm it here. As explained there, the confusion set of the DO-ADO method consists of signal-background pairs which are incorrectly ordered by the classifier trained at every step (called y_{pred} in Sec. II), with respect to the reference labels. When using truth labels for the latter, the confusion set can be significantly contaminated by signal-background pairs which may never be ordered properly, even by the ideal Neyman-Pearson classifier. This can in turn distort the ADO score which is calculated on the

[3] We note that in [29], the DO with truth labels was referred to as TO (for "truth-ordering") and it was pointed out that ADO with truth-labels reduces to the usual AUC metric.

confusion set. This explains why the LorentzNet-guided DO-ADO-FFS performs better than the truth-guided DO-ADO-FFS.

Meanwhile, we see from Fig. 4 that there is no significant difference in performance between truth-guided and LorentzNet-guided DisCo-FFS. This is perhaps the more expected and intuitive result. We believe the reason DisCo-FFS does not suffer from the degradation in performance when using truth labels can be understood by the fact that our confusion set is determined solely using the classifier trained at every step, and does not involve the reference labels at all. Also, our confusion set is determined on background and signal jets separately. Therefore, the issue of the forever-incorrectly-ordered signal-background pairs never even arises here. It would be interesting to test this explanation further, for example by combining these different ways of choosing the confusion set (DO or y_{pred}) with different relevance scores (ADO or DisCo). We reserve this for future work.

In any case, we conclude that, unlike DO-ADO-FFS, DisCo-FFS does not seem suffer in performance when using truth labels instead of a state-of-the-art pretrained tagger. This means that DisCo-FFS should be a suitable method for both *ab initio* feature selection and for explaining black box taggers.

D. Comparison with other taggers

The top-tagging comparison study [1] includes two methods which use high-level features as inputs for top-tagging; one used a NN with multibody N-subjettiness as input features [6,58], and the other uses a linear classification (with Fischer's linear discriminant) on EFPs. All other taggers are based on low-level jet information. The proposed DisCo-FFS selection strategy based on nine EFPs and three initial features outperforms all methods in the published study [1]. However, it falls short in performance to even more state-of-the-art taggers that were published after [1]: ParticleNet [14], LorentzNet [22], the ParT (particle transformer net, trained from scratch) tagger, ParT f.t.(particle transformer net, trained on the JetClass dataset [18], fine tuned on the landscape dataset) [18], and PELICAN [23]. Nevertheless, our tagger is able to achieve a very competitive performance with only 1440 parameters as shown in Table I and Fig. 5.

We also compare our performance to that of a network (architecture described in Appendix B) that was trained on all 7000 EFPs, along with m_J, p_T and $m_{W\text{−candidate}}$. As shown in Table I, this network is only able to obtain a performance of $R_{30} = 844$. This is significantly worse than the performance using the small subset of EFPs selected by DisCo-FFS. Clearly, the use of uninformative features in the training deteriorates the performance of the network. In principle, it should be possible to optimize the hyperparameters to recover the lost performance, but this is not so straightforward in practice, given the amount of time and resources it takes to train a network on all 7000 EFPs.[4] This emphasizes the need of doing feature selection.

TABLE I. AUC and R_{30} comparison of different taggers on the dataset from [1]. The R_{30} values of DisCo-FFS and ADO-FFS are the average R_{30}'s of ten classifier trainings, and the R_{30} of DNN on 7,000 EFPs is calculated over a single run. The performance for DisCo-FFS is after nine EFPs, whereas the performance reported for DO-ADO is after 17 EFPs.

Taggers	AUC	R_{30}	Param
Linear 1k EFPs [30]	0.980	384	1,000
N-sub 6 [6]	0.979	792 ± 18	57,000
N-sub 8 [6]	0.981	867 ± 15	58,000
ParticleNet [14]	0.986	1615 ± 93	366,000
ParticleNet-Lite [14]	0.984	1262 ± 49	26,000
LorentzNet [22]	0.987	2195 ± 173	224,000
ParT [18]	0.986	1602 ± 81	2,140,000
PELICAN [23]	0.987	2289 ± 204	45,000
DNN 7 k EFPs	0.980	844	237,000
DO-ADO (LorentzNet)	0.982	1212 ± 30	1,700
DisCo-FFS (truth)	0.982	1249 ± 43	1,400

As a further aside, this result also indicates why another popular feature selection method, which is based on assigning feature attributions using Shapley values, is not suitable here. Shapley values assume the existence of a high-performing classifier trained on a set of features, and then ranks those features in terms of their estimated contributions to the classifier outputs. In fact, the original Shapley values [42,43,46] are very much ill-suited to the problem at hand; their computational complexity grows exponentially with the number of features, so in practice can never be computed for more than ~10 features. Also the features are assumed to be uncorrelated, for the computation of Shapley values. With 7000 highly correlated features, this is clearly not the right approach. Later approaches such as SHAP [47] attempt to overcome the computational complexity issue by approximating the Shapley values in various ways. SHAP also used (approximate) Shapley values to unify different feature attribution methods [41,44,45,59] but generally all these works still assume independence of the features. This is an area of active research and it is possible a Shapley-inspired approach will work well on this problem in the future. Suffice to say that in our experiments (based on Deep SHAP [45,47] and the subpar DNN trained on 7000 EFPs), we obtained results that were only marginally better than random selection.

[4] This is also why the R_{30} quoted here does not come with an error bar from multiple retrainings; a single training was already prohibitively time consuming for us.

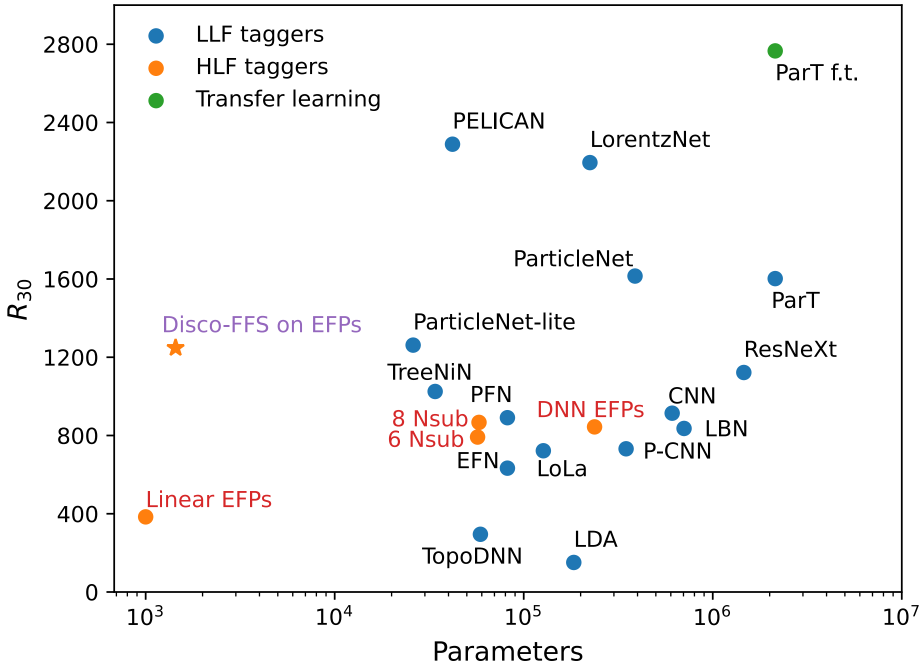


FIG. 5. R_{30} vs number of parameters of the model, for many different approaches to top-tagging. LorentzNet [22], PaticleNet [14], ParT, ParT f.t. [18], and PELICAN [23] are the some of the recent taggers with very good performances. “DisCo-FFS on EFPs” corresponds to the simple DNN trained on the first nine EFPs selected by DisCo-FFS, while “DNN EFPs” is our DNN trained on all the 7000 EFPs. The remaining taggers are taken from [1]. We see that the nine EFPs selected using DisCo-FFS have a very competitive performance, especially given the number of parameters.

E. Effectiveness of small training samples

To showcase another important benefit of feature selection, we compare the performance of the features we obtained using DisCo-FFS to ParticleNet and LorentzNet, on smaller training datasets. We take the set of features obtained in Sec. III C and train the same neural network with same hyperparameters on 5%, 1%, and 0.5% of the same training data. While both LorentzNet and ParticleNet had a superior performance for the full training dataset, our set of features outperforms ParticleNet at lower training fractions, and more-or-less matches LorentzNet at 0.5% and 1% of the training dataset, as shown in Fig. 6.

F. Robustness of the feature selection

It is interesting to ask whether the DisCo-FFS algorithm selects the same features every time. This is not *a priori* guaranteed, because there is some stochasticity to the algorithm, coming from the training of the NN classifier at every step (which in turn determines the confusion set on which the relevance score is calculated).

Shown in Fig. 7 is the R_{30} vs number of features selected, after running the DisCo-FFS algorithm five independent times. We see that DisCo-FFS repeatedly chooses the same first six EFPs. After that, the features selected start to diverge from fully deterministic, at first only slowly (there appear to be two possibilities for the pairs of EFPs selected in the 7th and 8th iterations), and then quickly from the 9th EFP onwards (on the 9th EFP, the five trials selected five different EFPs).

This is broadly consistent with Fig. 4. There we see the R_{30} shooting up rapidly during the first six EFPs, indicating that they provide a lot of classification power, and should produce a strong signal for the relevance score in the DisCo-FFS selection procedure. Then the R_{30} plateaus but does rise a little bit, from six EFPs to nine EFPs. This is consistent with a much weaker signal coming from the

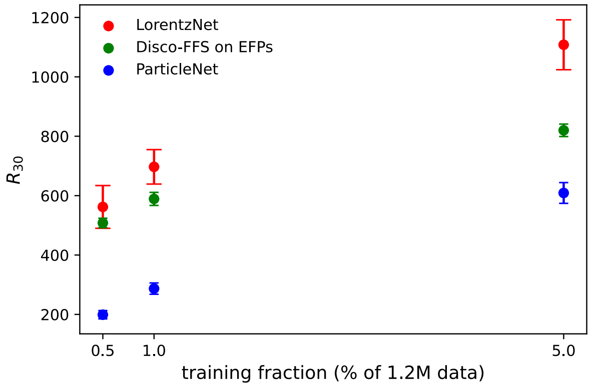


FIG. 6. Performance of training on 0.5%, 1%, and 5% of the training data. The EFPs selected using DisCo outperform ParticleNet, and match up to the performance of LorentzNet [22] at 0.5% of the total training data.

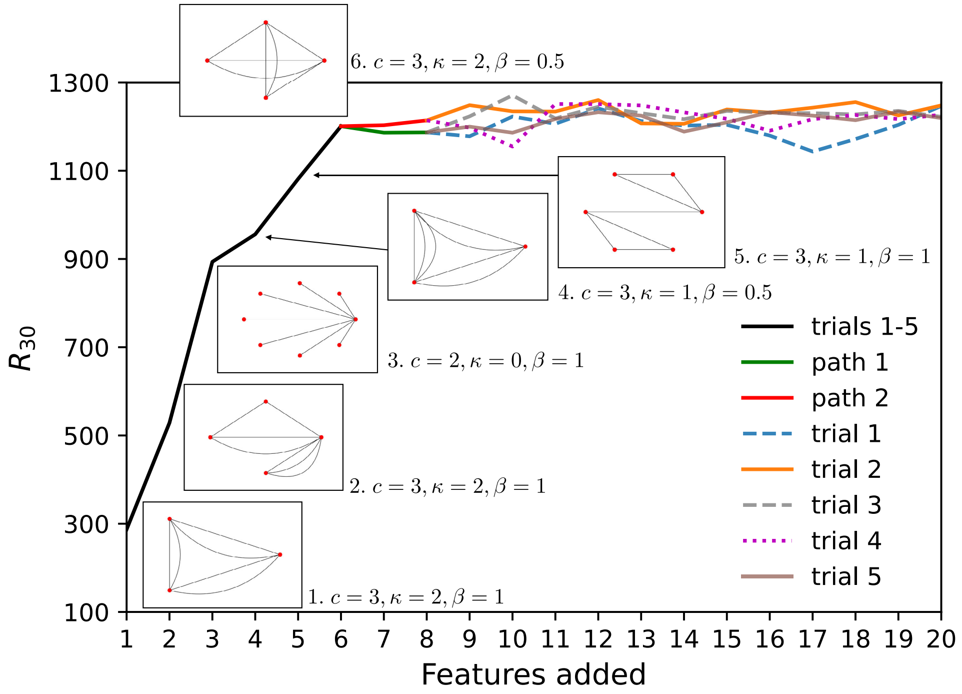


FIG. 7. Performance vs iteration for five trials of DisCo-FFS (performance is the mean R_{30} of ten trainings). We see that the feature selection is deterministic for the first six EFPs selected (superimposed), and there is a corresponding sharp rise in R_{30}. Then this is followed by two paths (marked path 1 and path 2) in the 7th and 8th iterations. After that, DisCo-FFS finds different sets of features to achieve similar performance.

relevance score and more possibility for randomness. Finally, after nine EFPs, the R_{30} no longer rises and instead fluctuates around 1250. This is consistent with the remaining EFPs being selected randomly and not providing any real signal to the relevance score.

G. Physical interpretation of the selected features

The selected energy flow polynomials can be used to gain physical insight for the case of top tagging. Shown in Tables II and III are the graphs, chromatic numbers c, (κ, β) values, and cumulative R_{30} values of the first eight EFPs selected by DisCo-FFS. We see that five of the first six EFPs selected are EFPs with $c = 3$. A chromatic number of a graph is the number of colors one can put to the nodes, so that no edges are connected by the same color. As noted in [30], the chromatic number of an EFP is also a proxy for the number of prongs in the jet. In other words, $c = 3$ EFPs are probes of 3-prong substructure—exactly what one would expect to be relevant for top tagging.

TABLE II. The EFPs selected by DisCo-FFS in the first six iterations.

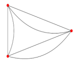
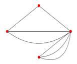
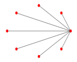
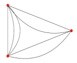
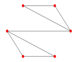
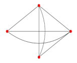
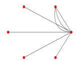
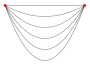
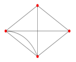
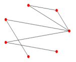
Iter	Feature	c	κ	β	R_{30}
1		3	2	1	287 ± 3
2		3	2	1	529 ± 10
3		2	0	1	894 ± 23
4		3	1	0.5	956 ± 35
5		3	1	1	1081 ± 22
6		3	2	0.5	1201 ± 23

TABLE III. Two paths selected by the EFPs in the 7th and 8th iteration.

	Iter	Feature	c	κ	β
Path 1:	7		2	0	0.5
	8		2	2	2

	Iter	Feature	c	κ	β
Path 2:	7		4	0.5	0.5
	8		3	1	1

Interestingly, there is one $c = 2$ EFP selected in the first six EFPs. This probe of 2-prong substructure could be related to the two prongs consisting of the b-quark and the boosted W-jet inside the top quark.

We also see from Table II that both IRC-safe and unsafe probes of 3-prong substructure are useful for tagging. The first two EFPs have $\kappa = 2$, and hence are an IRC-unsafe probe of hard radiation, with the first one being a 3-point correlator, and second one being a 4-point correlator.[5] IRC-safe EFPs ($\kappa = 1$) are not selected until the fourth and fifth iteration.

In the seventh and eighth iterations, there appear to be two possible paths for the FS algorithm to take, i.e., two unique possibilities for the pairs of EFPs selected. These are shown in Table III. In one of the paths, two IRC-unsafe EFPs probing the 2-prong substructure are selected with one of them probing small-angle radiation ($\beta = 0.5$), and the other one probing hard/wide-angle radiation ($\beta = 2$), which actually marks the first selected feature that probes wide-angle radiation. In the other path, we see the appearance of the first EFP which probes 4-prong substructure with small-angle radiation ($\beta = 0.5$), and this is followed up by an IRC-safe EFP probing 3-prong substructure.

Interestingly in our single run of `LorentzNet`-guided DisCo-FFS, the first six features are the same as Table II, whereas after that the 7th-EFP is the same one selected in Path 1 in III. This confirms that the similar performance between DisCo-FFS with truth and with `LorentzNet` is no coincidence, and is likely because `LorentzNet` (being so high performing) is quite close to the truth labels.

IV. CONCLUSIONS

In this work, we have introduced a new forward feature selection method, based on the distance correlation measure of statistical dependence—dubbed DisCo-FFS. Our method can operate equally well on either truth-labels (for *ab initio* feature selection) or on the outputs of a pretrained classifier (for explaining a black box AI).

We demonstrated the performance of our method using the task of boosted top tagging, as boosted top jets have a rich substructure and many subtle correlations that have proven to be a fruitful laboratory for developing increasingly powerful state-of-the-art taggers in the HEP literature.

Following [29], we have trained our DisCo-FFS method on a large set (7,000+) of energy flow polynomials, which aim to provide a complete description of the jet substructure. We have seen that DisCo-FFS is very effective at selecting EFPs from this large feature set; DisCo-FFS can achieve nearly-state-of-the-art top tagging performance (matching that of ParticleNet-lite [14]) with a selection of just a small number of EFPs (less than 10). We also show how it outperforms the DO-ADO-FFS method of [29] (which we have attempted to replicate as closely as possible), consistently achieving higher tagging performance after each EFP that is selected.

The fact that our method falls short of the most state of the art deep learning methods (ParT [18], PELICAN [23], and LorentzNet [22]) is interesting. Either our method is not fully optimal at selecting the features, or the 7,000+ EFPs we used as the basis of our study do not capture all the physics underlying top tagging. A possible follow-up study to further probe this question would be to supplement the 7,000+ EFPs with additional jet substructure variables, for instance the subjettiness variables of [58,60], jet spectra and morphological features of [61–63], or boost invariant polynomials [64]. This observation also raises the possibility that there might be more meaningful jet substructure variables out there, beyond those that are presently known, waiting to be discovered. This is obviously an interesting avenue for future research.

Beyond simple object tagging, DisCo-FFS might also be able to shine for tasks—such as building supervised classifiers for new physics discovery—where calibration of the algorithm is difficult and a small number of well-understood features is preferable. While particle physics is in an especially good position due to the presence of well-motivated bases of features (such as the used EFPs) such decompositions also exists for other domains, e.g., in the forms of wavelets applied to images (e.g., building on [65]).

In general, EFPs selected could make for a very lightweight and performant top tagger. This could have important applications to triggering [66]. For that, a fast way to calculate EFPs on FPGAs would be required. Such will be interesting to explore further.

It would also be potentially illuminating to study the robustness of the selected EFPs under domain shift. For example, recently ATLAS released an official top-tagging dataset [67]. One could compare the EFPs selected by DisCo-FFS on the different top tagging datasets, and see how one set of EFPs performs on the other dataset. One could also imagine training this method on a restricted set of HLFs (EFPs or otherwise) that are deemed to be well-modeled by simulations. This could help with the calibration and robustness of taggers developed using simulation and deployed on data.

Overall, we observe the start of a positive feedback loop between deep learning method development and physics-motivated feature discovery. Each one drives the other. Early top taggers [68] started with jet substructure variables like N-subjettiness. Then it looked like deep learning was able to go way beyond HLFs and we would have to rely on fully automated feature engineering. Now there is some signs that we are coming full circle. Ultimately we may hope to match the performance of the SOTA deep learning taggers with just a handful of (yet-to-be-invented?) HLFs.

[5] We emphasize that all the HLFs we use in this work are actually IRC-safe in the end, since they are constructed from detector-reconstructed particles.

This would be a very satisfying outcome, proving that deep learning does not have to be a black box but can drive fundamental physics discoveries.

The code for this paper can be found at [69].

ACKNOWLEDGMENTS

We are grateful to Taylor Faucett, Daniel Whiteson and especially Jesse Thaler for discussions and help regarding the W-jets dataset, the DO-ADO method, and truth vs black-box guiding. We are also grateful to Sitian Qian for assisting us with the `LorentzNet` classifier output. We thank Purvasha Chakravarti and Jose M. Muñoz Arias for helpful discussions. Finally, we thank Daniel Whiteson for comments on the draft. G. K. acknowledges support by the Deutsche Forschungsgemeinschaft under Germany's Excellence Strategy—EXC 2121 Quantum Universe—390833306. The work of R. D. and D. S. was supported by DOE Grant No. DOE-SC0010008. The authors acknowledge the Office of Advanced Research Computing (OARC) at Rutgers, The State University of New Jersey [70] for providing access to the Amarel cluster and associated research computing resources that have contributed to the results reported here.

APPENDIX A: VALIDATION OF OUR IMPLEMENTATION OF DO-ADO-FFS

1. The DO-ADO feature selection method

In this appendix, we validate our implementation of the DO-ADO feature selection method of [29]. This method is based on the DO and ADO metrics, which we will now explain.

For a signal event x_s and a background event x_b, the DO metric is given by

$$\mathrm{DO}(x_s, x_b; y_{\mathrm{pred}}, y_{\mathrm{ref}}) = \Theta((y_{\mathrm{pred}}(x_s) - y_{\mathrm{pred}}(x_b)) \times (y_{\mathrm{ref}}(x_s) - y_{\mathrm{ref}}(x_b))), \tag{A1}$$

where Θ is the Heaviside step function. In other words, $\mathrm{DO} = 1$ ($\mathrm{DO} = 0$) if the pair of events has the same (different) ordering under y_{pred} as under the reference classifier y_{ref}.

Meanwhile, average decision ordering is defined over a dataset $\mathcal{D}$ consisting of pairs of signal and background events,

$$\mathrm{ADO}(\mathcal{D}; y_{\mathrm{pred}}, y_{\mathrm{ref}}) = \langle \mathrm{DO}(x_s, x_b; y_{\mathrm{pred}}, y_{\mathrm{ref}}) \rangle_{(x_s, x_b) \sim \mathcal{D}}. \tag{A2}$$

In other words, ADO is the average of the DO metric over the dataset.

The DO-ADO feature selection algorithm [29] also follows the same steps 1 and 4, as described in Sec. II. For steps 2 and 3, we have

Step 2: The confusion set X_0 is formed out of pairs of (signal, background) events with $\mathrm{DO}(x_s, x_b; y_{\mathrm{pred}}, y_{\mathrm{ref}}) = 0$. It is too computationally intensive to find and analyze all possible pairs of events with $\mathrm{DO} = 0$, so only a randomly selected subset of (signal, background) pairs is considered for X_0.

Step 3: The relevance score for each feature f is defined as

$$s_f = \mathrm{ADO}(X_0; f, y_{\mathrm{ref}}). \tag{A3}$$

So a feature with a larger ADO value would be one for which more events in the confusion set are correctly ordered by the feature. The idea of DO-ADO-FFS is to identify the feature at every step that most correctly orders signal vs background events that are incorrectly ordered by the previous step, with respect to the reference classifier y_{ref}.

2. Validation with W-tagging

To validate our implementation of DO-ADO-FFS, we train it on the same W-tagging dataset considered in [29] with respect to truth labels,[6] and demonstrate that we achieve the same performance as shown there.

As in [29], we start with an initial feature set of

$$\mathcal{F}_{\mathrm{initial}} = \mathcal{F}_2 = \{m_J, p_T\}. \tag{A4}$$

Here we apply both truth-guided DO-ADO-FFS and DisCo-FFS to the same set of EFPs considered in [29] and this paper. The results (as AUC and R_{50} vs number of features selected) are shown in Fig. 8, together with the performance metrics for a reference CNN tagger from [29], as well as the reference AUC value of 0.951, at which the truth-guided ADO in [29] was mentioned to saturate after seven features.

For the ADO method, we see that the AUC reaches around 0.951 after seven features. This matches the description in [29] and demonstrates that we have successfully validated the implementation of DO-ADO-FFS. Interestingly, however, we notice that the AUC of our version saturates at a slightly higher AUC of around 0.952.

Meanwhile, DisCo-FFS again outperforms DO-ADO-FFS; it reaches the CNN AUC after eight features, and actually proceeds to exceed the performance of the CNN—all without using any knowledge of the CNN classifier

[6] We could not perform DO-ADO-FFS with respect to the pretrained CNN because this was not made publicly available at the time of this publication.

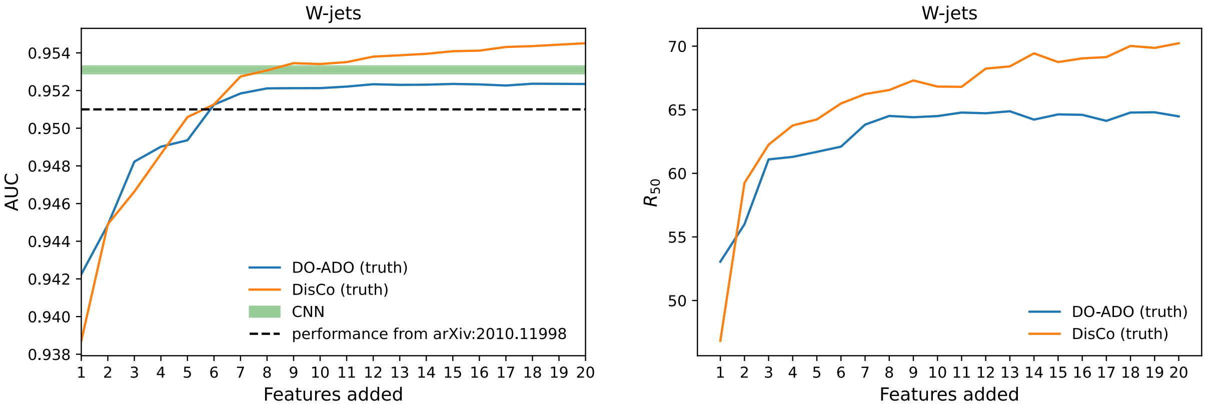


FIG. 8. Left: AUC vs number of features selected, for DO-ADO (blue) and DisCo (orange), both truth-guided. The green line indicates the AUC of the reference CNN tagger from [29], while the black dashed line indicates the performance that truth-guided DO-ADO achieved in [29]. Here we see our version of the truth-guided DO-ADO method saturates at a slightly higher AUC of 0.952 (but still short of the CNN AUC), whereas the DisCo-FFS method reaches the CNN AUC after eight features, and is able to exceed the CNN AUC. Right: Same comparison but in terms of the R_{50} (rejection power at 50% true positive rate) metric.

output! This shows the potential promise of a well-designed forward feature selection method operating on a well-chosen feature set; it could conceivably show that a deep learning classifier is not actually as state-of-the-art as previously thought.

APPENDIX B: HYPERPARAMETERS AND ARCHITECTURES

For our feature set, we use $\log(\mathrm{EFP} + 10^{-40})$, instead of the bare EFPs as our features, during training, as well as during feature selection, and we see that this leads to a better performance.

Due to computational constraints, we actually calculate DisCo using minibatches. We divide the confusion set X_0 into minibatch sizes of 2048, and then average over all the minibatches to estimate DisCo over the confusion set.

TensorFlow was used for training classifiers for DisCo-FFS and DO-ADO-FFS, and the following hyperparameters were used:

(i) Two hidden layers of 16 nodes with ReLU activation, final output layer with `softmax` activation;

(ii) A `RobustScaler` is fitted on the training and validation data combined and is used to rescale the dataset;

(iii) We use the `Adam` optimizer with default hyperparameters for 500 epochs, with minibatch size = 512;

(iv) Model checkpoint is used to save the model with the minimum validation loss.

We observed that the final R_{30}'s were higher after the use of a slightly bigger network with 32×32 hidden layers, so we retrained all the features (after the FFS) with this network, and obtained our final R_{30}'s, including Fig. 4, with this network.

The DNN trained on all 7000 EFPs uses the same hyperparameters as discussed above, but we use a slightly bigger network with three hidden layers of 32 nodes.

For both the truth-guided DisCo-FFS and DO-ADO methods, we apply feature selection to the combined training and validation sets. However, for the `LorentzNet`-guided versions, we apply the feature selection only to the validation set. This is because we noticed a significant overfitting of `LorentzNet` to the training set, as compared to the validation and the test set.

APPENDIX C: AFFINE-INVARIANT DISTANCE CORRELATION

Distance correlation is a correlation metric which can quantify nonlinear correlations in the joint distribution of two random vectors $(\vec{X}, \vec{Y})$ of arbitrary dimension [34–37]. In particular, DisCo is zero if and only if $\vec{X}$ and $\vec{Y}$ are statistically independent $[p(\vec{X}, \vec{Y}) = p(\vec{X})p(\vec{Y})]$, and positive otherwise.

With $\vec{X}$ and $\vec{Y}$ as 1D vectors, DisCo has used been previously used in physics for decorrelation of neural networks against mass [38]. However, DisCo is even more powerful than that—it can also measure statistical dependence of multivariate distributions, a powerful property that enables the forward feature selection algorithm described in this work.

For our case, $\vec{X} = y_{\mathrm{truth}}$ is a 1D vector, and $\vec{Y} = (f_{i_1}, f_{i_2}, \ldots, f_{i_n})$ is an n-dimensional feature vector. The population value of squared distance covariance of $\vec{X}$ and $\vec{Y}$ is given by

$$\begin{aligned}\mathrm{dCov}^2(\vec{X},\vec{Y}) \coloneqq{}& E[\|\vec{X}-\vec{X}'\|\|\vec{Y}-\vec{Y}'\|] \\ &+ E[\|\vec{X}-\vec{X}'\|]E[\|\vec{Y}-\vec{Y}'\|] \\ &- 2E[\|\vec{X}-\vec{X}'\|\|\vec{Y}-\vec{Y}''\|], \end{aligned} \tag{C1}$$

where $(\vec{X},\vec{Y})$, $(\vec{X}',\vec{Y}')$, and $(\vec{X}'',\vec{Y}'')$ are independent and identically distributed from the distribution $(\vec{X},\vec{Y})$ and $\|.\|$ is the Euclidean vector norm.

Distance correlation is given by

$$\mathrm{dCor}^2(\vec{X},\vec{Y}) = \frac{\mathrm{dCov}^2(\vec{X},\vec{Y})}{\sqrt{\mathrm{dCov}^2(\vec{X},\vec{X})\mathrm{dCov}^2(\vec{Y},\vec{Y})}}, \tag{C2}$$

which is normalized between 0 and 1.

Finally, using the covariance matrices Σ_X, Σ_Y, affine-invariant distance correlation is simply

$$\overline{\mathrm{dCor}}^2(\vec{X},\vec{Y}) = \mathrm{dCor}^2(\Sigma_X^{-1/2}\vec{X}, \Sigma_Y^{-1/2}\vec{Y}). \tag{C3}$$

In this work, we use the DCOR package [71] for the computation of distance correlation and affine-invariant distance correlation.

[1] G. Kasieczka *et al.*, The machine learning landscape of top taggers, SciPost Phys. **7**, 014 (2019).

[2] J. Shlomi, P. Battaglia, and J.-R. Vlimant, Graph neural networks in particle physics, Mach. Learn. **2**, 021001 (2021).

[3] G. Karagiorgi, G. Kasieczka, S. Kravitz, B. Nachman, and D. Shih, Machine learning in the search for new fundamental physics, arXiv:2112.03769.

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

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

[6] L. Moore, K. Nordström, S. Varma, and M. Fairbairn, Reports of my demise are greatly exaggerated: *N*-subjettiness taggers take on jet images, SciPost Phys. **7**, 036 (2019).

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

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

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

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

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

[12] S. Macaluso, Recursive neural network for jet physics, https://github.com/SebastianMacaluso/RecNN_TensorFlow (2018).

[13] P. T. Komiske, E. M. Metodiev, and J. Thaler, Energy flow networks: Deep sets for particle jets, J. High Energy Phys. 01 (2018) 121.

[14] H. Qu and L. Gouskos, Jet tagging via particle clouds, Phys. Rev. D **101**, 056019 (2020).

[15] F. A. Dreyer and H. Qu, Jet tagging in the Lund plane with graph networks, J. High Energy Phys. 03 (2021) 052.

[16] E. A. Moreno, O. Cerri, J. M. Duarte, H. B. Newman, T. Q. Nguyen, A. Periwal, M. Pierini, A. Serikova, M. Spiropulu, and J.-R. Vlimant, JEDI-net: A jet identification algorithm based on interaction networks, Eur. Phys. J. C **80**, 58 (2020).

[17] V. Mikuni and F. Canelli, Point cloud transformers applied to collider physics, Mach. Learn. Sci. Tech. **2**, 035027 (2021).

[18] H. Qu, C. Li, and S. Qian, Particle transformer for jet tagging, in *Proceedings of the 39th International Conference on Machine Learning* (2022), arXiv:2202.03772.

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

[20] M. Erdmann, E. Geiser, Y. Rath, and M. Rieger, Lorentz boost networks: Autonomous physics-inspired feature engineering, J. Instrum. **14**, P06006 (2018).

[21] A. Bogatskiy, B. Anderson, J. T. Offermann, M. Roussi, D. W. Miller, and R. Kondor, Lorentz group equivariant neural network for particle physics, arXiv:2006.04780.

[22] S. Gong, Q. Meng, J. Zhang, H. Qu, C. Li, S. Qian, W. Du, Z.-M. Ma, and T.-Y. Liu, An efficient lorentz equivariant graph neural network for jet tagging, J. High Energy Phys. 07 (2022) 030.

[23] A. Bogatskiy, T. Hoffman, D. W. Miller, and J. T. Offermann, Pelican: Permutation equivariant and Lorentz invariant or covariant aggregator network for particle physics, arXiv:2211.00454.

[24] A. Khot, M. S. Neubauer, and A. Roy, A detailed study of interpretability of deep neural network based top taggers, Mach. Learn. Sci. Tech. **4**, 035003 (2023).

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

[26] S. Diefenbacher, H. Frost, G. Kasieczka, T. Plehn, and J. Thompson, CapsNets continuing the convolutional quest, SciPost Phys. **8**, 023 (2020).

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

[28] G. Agarwal, L. Hay, I. Iashvili, B. Mannix, C. McLean, M. Morris, S. Rappoccio, and U. Schubert, Explainable AI for ML jet taggers using expert variables and layerwise relevance propagation, J. High Energy Phys. 05 (2021) 208.

[29] T. Faucett, J. Thaler, and D. Whiteson, Mapping machine-learned physics into a human-readable space, Phys. Rev. D **103**, 036020 (2021).

[30] P. T. Komiske, E. M. Metodiev, and J. Thaler, Energy flow polynomials: A complete linear basis for jet substructure, J. High Energy Phys. 04 (2018) 013.

[31] J. Collado, K. Bauer, E. Witkowski, T. Faucett, D. Whiteson, and P. Baldi, Learning to isolate muons, J. High Energy Phys. 10 (2021) 200.

[32] J. Collado, J. N. Howard, T. Faucett, T. Tong, P. Baldi, and D. Whiteson, Learning to identify electrons, Phys. Rev. D **103**, 116028 (2021).

[33] T. Faucett, S.-C. Hsu, and D. Whiteson, Learning to identify semi-visible jets, J. High Energy Phys. 12 (2022) 132.

[34] G. J. Székely and M. L. Rizzo, Brownian distance covariance, Ann. Appl. Stat. **3**, 1236 (2009).

[35] G. J. Székely, M. L. Rizzo, and N. K. Bakirov, Measuring and testing dependence by correlation of distances, Ann. Stat. **35**, 2769 (2007).

[36] G. J. Székely and M. L. Rizzo, Partial distance correlation with methods for dissimilarities, Ann. Stat. **42**, 2382 (2014), arXiv:1310.2926.

[37] G. J. Székely and M. L. Rizzo, The distance correlation t-test of independence in high dimension, J. Multivariate Anal. **117**, 193 (2013).

[38] G. Kasieczka and D. Shih, Robust jet classifiers through distance correlation, Phys. Rev. Lett. **125**, 122001 (2020).

[39] G. Kasieczka, B. Nachman, M. D. Schwartz, and D. Shih, Automating the ABCD method with machine learning, Phys. Rev. D **103**, 035021 (2021).

[40] V. Mikuni, B. Nachman, and D. Shih, Online-compatible unsupervised nonresonant anomaly detection, Phys. Rev. D **105**, 055006 (2022),

[41] S. Lipovetsky and M. Conklin, Analysis of regression in game theory approach (2001), https://onlinelibrary.wiley.com/doi/pdf/10.1002/asmb.446.

[42] *The Shapley Value: Essays in Honor of Lloyd S. Shapley* (Cambridge University Press, Cambridge, England, 1988).

[43] S. Hart, Shapley value, in *Game Theory*, edited by J. Eatwell, M. Milgate, and P. Newman (Palgrave Macmillan, UK, London, 1989), pp. 210–216.

[44] E. Štrumbelj and I. Kononenko, Explaining prediction models and individual predictions with feature contributions, Knowl. Inf. Syst. **41**, 647 (2013).

[45] A. Shrikumar, P. Greenside, and A. Kundaje, Learning important features through propagating activation differences, arXiv:1704.02685.

[46] L. S. Shapley, 17. A value for n-person games, in *Contributions to the Theory of Games (AM-28), Volume II*, edited by H. W. Kuhn and A. W. Tucker (Princeton University Press, Princeton, NJ, 2016), pp. 307–318.

[47] S. M. Lundberg and S.-I. Lee, A unified approach to interpreting model predictions, in *Advances in Neural Information Processing Systems 30*, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Curran Associates, Inc., Red Hook, NY, 2017), pp. 4765–4774.

[48] E. Song, B. L. Nelson, and J. Staum, Shapley effects for global sensitivity analysis: Theory and computation, SIAM/ASA J. Uncertainty Quantif. **4**, 1060 (2016).

[49] K. Aas, M. Jullum, and A. Løland, Explaining individual predictions when features are dependent: More accurate approximations to shapley values, Artif. Intell. **298**, 103502 (2021).

[50] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee, Explainable AI for trees: From local explanations to global understanding, arXiv:1905.04610.

[51] N. Sellereite and M. Jullum, shapr: An r-package for explaining machine learning models with dependence-aware shapley values, J. Open Source Softwaare **5**, 2027 (2019).

[52] J. Dueck, D. Edelmann, T. Gneiting, and D. Richards, The affinely invariant distance correlation, Bernoulli **20**, 2305 (2014).

[53] G. Kasieczka, T. Plehn, J. Thompson, and M. Russel, Top quark tagging reference dataset, 10.5281/zenodo.2603256 (2019).

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

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

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

[57] M. Cacciari, G. P. Salam, and G. Soyez, The anti-kt jet clustering algorithm, J. High Energy Phys. 04 (2008) 063.

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

[59] M. T. Ribeiro, S. Singh, and C. Guestrin, "why should I trust you?": Explaining the predictions of any classifier, arXiv:1602.04938.

[60] K. Datta, A. Larkoski, and B. Nachman, Automating the construction of jet observables with machine learning, Phys. Rev. D **100**, 095016 (2019).

[61] A. Chakraborty, S. H. Lim, and M. M. Nojiri, Interpretable deep learning for two-prong jet classification with jet spectra, J. High Energy Phys. 19 (2019) 135.

[62] A. Chakraborty, S. H. Lim, M. M. Nojiri, and M. Takeuchi, Neural network-based top tagger with two-point energy

correlations and geometry of soft emissions, J. High Energy Phys. 07 (2020) 111.

[63] S. H. Lim and M. M. Nojiri, Morphology for jet classification, Phys. Rev. D **105**, 014004 (2022).

[64] J. M. Munoz, I. Batatia, and C. Ortner, Bip: Boost invariant polynomials for efficient jet tagging, Mach. Learn. Sci. Tech. **3**, 04LT05 (2022).

[65] V. Rentala, W. Shepherd, and T. M. P. Tait, Tagging boosted Ws with wavelets, J. High Energy Phys. 08 (2014) 042.

[66] J. Duarte *et al.*, Fast inference of deep neural networks in FPGAs for particle physics, J. Instrum. **13**, P07027 (2018).

[67] ATLAS Collaboration, Constituent-based top-quark tagging with the ATLAS detector, Technical Report, CERN, Geneva, 2022, all figures including auxiliary figures are available at https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PUBNOTES/ATL-PHYS-PUB-2022-039.

[68] D. Adams *et al.*, Towards an understanding of the correlations in jet substructure, Eur. Phys. J. C **75**, 409 (2015).

[69] https://github.com/rd804/DisCo-FFS

[70] https://it.rutgers.edu/oarc.

[71] DCOR, https://dcor.readthedocs.io/en/latest/index.html.