Eur. Phys. J. C (2022) 82:1162
https://doi.org/10.1140/epjc/s10052-022-11084-4

THE EUROPEAN PHYSICAL JOURNAL C

Regular Article - Theoretical Physics

Exploring the universality of hadronic jet classification

Kingman Cheung[1,5,a], **Yi-Lun Chung**[1,b], **Shih-Chieh Hsu**[2,c], **Benjamin Nachman**[3,4,d]

[1] Department of Physics and Center for Theory and Computation, National Tsing Hua University, Hsinchu 300, Taiwan
[2] Department of Physics, University of Washington, Seattle, WA 98195, USA
[3] Physics Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA
[4] Berkeley Institute for Data Science, University of California, Berkeley, CA 94720, USA
[5] Division of Quantum Phases and Devices,School of Physics, Konkuk University, Seoul 143-701, Republic of Korea

Received: 21 April 2022 / Accepted: 29 November 2022 / Published online: 22 December 2022

Abstract The modeling of jet substructure significantly differs between Parton Shower Monte Carlo (PSMC) programs. Despite this, we observe that machine learning classifiers trained on different PSMCs learn nearly the same function. This means that when these classifiers are applied to the same PSMC for testing, they result in nearly the same performance. This classifier universality indicates that a machine learning model trained on one simulation and tested on another simulation (or data) will likely be optimal. Our observations are based on detailed studies of shallow and deep neural networks applied to simulated Lorentz boosted Higgs jet tagging at the LHC.

Contents

1 Introduction

Deep learning is becoming widely used for various classification tasks in collider physics (see e.g., Refs. [1–6]). One of the core benefits of deep learning over traditional analysis techniques is that it is able to identify patterns in very high-dimensional feature spaces. At the Large Hadron Collider (LHC), such low-level inputs are dominated by hadronic activity. Most machine learning approaches are trained using Parton Shower Monte Carlo (PSMC) simulations that produce exclusive final states with the same complexity as real data [7]. However, there are significant variations between PSMCs due to the large number of perturbative and non-perturbative modeling assumptions.

These variations lead to potential biases and suboptimal sensitivity in data analyses [8]. A bias occurs when the simulation model used for inference (given an analysis strategy) is not the same as nature. There is a large and growing literature on methods to reduce biases from PSMC model variations through decorrelation [9–12] and other approaches [13–15]. A key challenge with modeling uncertainties in contrast to experimental uncertainties is that they are often estimated by comparing two simulations. This difference does not have a statistical origin and may not be the full uncertainty, so caution is required to reduce the uncertainty through automated approaches [16]. A general solution to estimating (and then reducing) systematic uncertainties from PSMC variations is still an active area of research and development.[1]

In principle, the same challenge exists when quantifying suboptimal performance due to PSMC variations. Suboptimal performance occurs when the simulation model used for training a machine learning model is different than nature.

[a] e-mail: cheung@phys.nthu.edu.tw
[b] e-mail: s107022801@m107.nthu.edu.tw (corresponding author)
[c] e-mail: schsu@uw.edu
[d] e-mail: bpnachman@lbl.gov

[1] See Refs. [17,18] for the possibility of using machine learning to bound these uncertainties.

While not directly a source of systematic uncertainty, this suboptimality has important consequences for the physics program of the LHC. To quantify the suboptimality, one could compare different PSMC models, as is done for determine the systematic uncertainty. This has the same unsatisfying properties as described above.

However, there have been a number of hints in the literature that the suboptimality due to PSMC variations may actually be small. For example, Ref. [19] observed that training a quark versus gluon jet classifier with the HERWIG [20] PSMC and then applying it to jets simulated with the PYTHIA [21] PSMC has nearly the same performance as training with PYTHIA and also testing on PYTHIA (with a statistically identical, but independent dataset). This small difference in performance is contrasted to the large difference in performance when testing on jets from HERWIG. A similar result was observed in the context of signal jets in Ref. [22]. From these observations, we conjecture that the deep learning models are learning universal properties of quantum chromodynamics (QCD). We hypothesis that the performance gaps present when the test sets differ simply reflects variations in the amount of QCD radiation, but not the type of information that is useful for discrimination.

To build intuition for this conjecture, consider the case of quark versus gluon jet tagging. At leading logarithmic (LL) order and considering only infrared and collinear safe observables, the optimal classifier is simply iterated-soft-drop multiplicity inside the jet [23]. This statement is true independent of the strong coupling constant, α_s. However, common metrics of performance such as the Area Under the Curve (AUC) depend on α_s;[2] when there are more emissions (higher α_s), the quark and gluon perturbative multiplicity distributions are more separable. In particular, at LL, perturbative multiplicity is a Poisson random variable with a mean that is proportional to a color factor multiplied by α_s. As α_s grows, the gluon distribution grows significantly faster than the quark one:

$$\frac{\mu_g - \mu_q}{\sqrt{\sigma_g^2 + \sigma_q^2}} \sim \frac{\alpha_s(C_F - C_A)}{\sqrt{\alpha_s C_F + \alpha_s C_A}} \propto \sqrt{\alpha_s}\,, \tag{1.1}$$

where $C_F = 4/3$ ($C_A = 3$) is the quark (gluon) color factor. Imagine that two PSMCs had the same physics approximations, but different values of α_s. They would find the same classifier and thus if the test set is the same, the performance would be the same.

Our goal is to test the universality hypothesis in detail using the important benchmark problem of Lorentz boosted Higgs boson jet versus QCD jet tagging. In this context, universality means that the learned classifiers are the same up to a monotonic re-scaling, which means that they result in the same decision boundaries. We consider both shallow and deep learning models as well as a variety of PSMC models.

This paper is organized as follows. A concrete example are introduced in Sect. 2. Architectures of deep-learning classifiers are in Sect. 3. The results are provided in Sect. 4. The paper ends with conclusions and outlook in Sect. 5.

2 Numerical examples

Lorenz-boosted Higgs tagging, focusing on the $b\bar{b}$ final state, is the example in this study. High-level features and low-level inputs are used to train shallow and deep-learning classifiers.

2.1 Monte Carlo samples

This study considers Lorenz-boosted Higgs tagging, focusing on the $b\bar{b}$ final state. The signal is high p_T Higgs bosons and the background is generic quark and gluon jets. The hard-scatter reactions are common to all parton shower models and are generated with MadGraph5_aMC@NLO 2.7.3 [24] for modeling pp collisions at $\sqrt{s}$ = 14 TeV. The PDF4LHC15_nnlo_mc [25] parton distribution function and the NNPDF30_nlo_as_0118 [26] parton distribution function are used for signal and background, respectively.

The hard-scattering events are passed to PYTHIA 8.303 [21] to simulate the parton shower, using three different complete parton-shower frameworks. The first one is default setting, where evolution variable is virtuality of the off-shell propagator. The second framework is Virtual Numerical Collider with Interleaved Antennae (VINCIA) shower [27–29], where the evolution variable is transverse momentum for QCD + EW/QED showers based on the antenna formalism. The last framework is Dipole resummation (DIRE), which is a transverse-momentum ordered dipole shower. The PYTHIA family uses the string model [30,31] for hadronization. The string model is based on string fragmentation function to break string to form hardons. HERWIG 7.2.2 [20] with angularly-ordered showers is also used to model the parton shower. The cluster model [32,33] is implemented in HERWIG 7.2.2. The cluster hadronization model is based on preconfinement. This model forcibly decays gluons into quark-antiquark pairs and form neutral clusters. PYJET [34,35] and the anti-k_t [36] algorithm with radius parameter R = 1.0 are used to define the jets.

An event preselection similar to Ref. [37] is used to reject most background events. The Higgs-like jet is required to satisfy 300 GeV $< p_T^J <$ 500 GeV, 110 GeV $<$ invariant mass of the jet (M_J) $<$ 160 GeV and to be double b-tagged. Jets are declared double b-tagged if they have two or more ghosted-associated [38,39] B hadrons. After the preselec-

[2] If Casimir scaling were holding the AUC would have been independent of α_s. However, multiplicity breaks the Casimir scaling such that the AUC depends on α_s.

tion, the high-level jet features and low-level features are used to probe the universality of discriminating boosted Higgs jets from QCD jets.

Since the goal of this paper is to investigate the universality of hadronic jet classification, there are a number of simplifying assumptions. The background in the study is only generic quark and gluon jets. The relatively smaller $t\bar{t}$ background is ignored. For each PSMC setup, the default parameters are used.

2.2 High-level features

In order to distinguish Higgs jets via Gradient Tree Boosting (BDT) and a fully connected / dense neural network, the following six commonly-used high-level features are considered:

1. M_J: invariant mass of the leading jet;
2. $\tau_{21} = \tau_2/\tau_1$: n-subjettiness ratio [40,41];
3. $D_2^{(\beta)} = e_3^{(\beta)}/(e_2^{(\beta)})^3$ with $\beta = 1, 2$: energy correlation function ratios [42];
4. $C_2^{(\beta)} = e_3^{(\beta)}/(e_2^{(\beta)})^2$ with $\beta = 1, 2$: energy correlation function ratios [43];

where e_i is the normalized sum over doublets ($i = 2$) or triplets ($i = 3$) of constituents inside jets, weighted by the product of the constituent transverse momenta and pairwise angular distances. For this analysis, β is considered to be 1 and 2.

The distributions of these six variables are shown in Fig. 1, in which the capability of each observable to discriminate between signal and background is demonstrated. The salient features of these histograms are described below.

The jet invariant mass distribution peaks near the Higgs boson mass of 125 GeV [44] for the signal and has a broad distribution for the background. In the setup of this study, Herwig 7.2.2 with angularly-ordered showers leads to slightly higher and broader signal peak due to different underlying event structure compared to Pythia 8.303. Similarly, the distributions of τ_{21}, D_2^{β}, and C_2^{β} show similar position and shape of the peak among the Pythia PSMC's, but somewhat different for the Herwig Angular. The two-prong structure due to the decay of massive objects into two hard QCD partons in the case of the signal jets results in low τ_{21}, D_2 and C_2.

2.3 Low-level features

The low-level inputs to the CNN are images of Higgs-like jet [45,46]. The resolution is 40×40 pixels and in 1R×1R range, where R is the jet radius. The images consist of three channels, analogous to the Red-Green-Blue (RGB) channels of a color image [19]. The pixel intensity for the three channels correspond to the sum of the charged particle p_T, the sum of the neutral particle p_T, and the number of charged particles in a given region of the image. The Higgs-like jet images are rotated to align along two-subject's axis. The leading subjet is at the origin and the subleading subjet is directly below the leading subjet. If there is a third-leading subjet, the image will be reflected. All images are normalized so that the intensities all sum to unity.[3] After normalization, the pixel intensities are standardized so that their distribution has mean zero and unit variance. Figure 2 shows the average Higgs-like jet images in the charged p_T channel. The patterns in the charged p_T channel are similar to the other two channels.

Figure 3 shows the difference between the four PSMC algorithms with respect to Pythia 8.303 default showering, referred to as the nominal simulation. The substructure in jets are different among the other three PSMC simulations with respect to the nominal sample due to different approximations made in the final state radiation and other QCD effects. This diversity of the PSMC approaches may effect the performance of jet classifiers trained on low-level features. Therefore, we train a convolutional neural network-based jet classifier to explore this generator-dependence of classification performance.

3 Classifier architectures

The BDT has a fixed number of estimators (1400) with maximum depth 5. The minimum number of samples is fixed at 5% as required to split an internal node and 1% as required to be at a leaf node. This BDT model is trained on the high-level features of the jet using the `scikit-learn` library [48]. KerasTuner [49] is used to get the best configuration of hyperparameters.

The dense neural network has four full connected layers. There are 224, 928, 288 and 1024 neurons, respectively. Rectified linear unit (ReLU) activation functions are used for all layers of this neural network. Before the output layer, Dropout [50] regularization is added to reduce overfitting with a dropout rate = 0.01. For this two-class problem, the activation function of the output layer is a sigmoid function. The binary cross entropy loss function is optimized during the training. The Adam optimizer [51] with a learning rate of 6.5428×10^{-5} is used to select the network weights. The KerasTuner [49] is used to get the best configuration of hyperparameters. The `Keras-2.4.0` library is used to train the dense neural network models with the `TensorFlow-2.4.1` [52] backend, on a `NVIDIA A100 SXM 80GB` Graphical Processing Unit (GPU).

[3] This may remove useful discriminating information; however, it significantly improves the stability of the machine learning training [47].

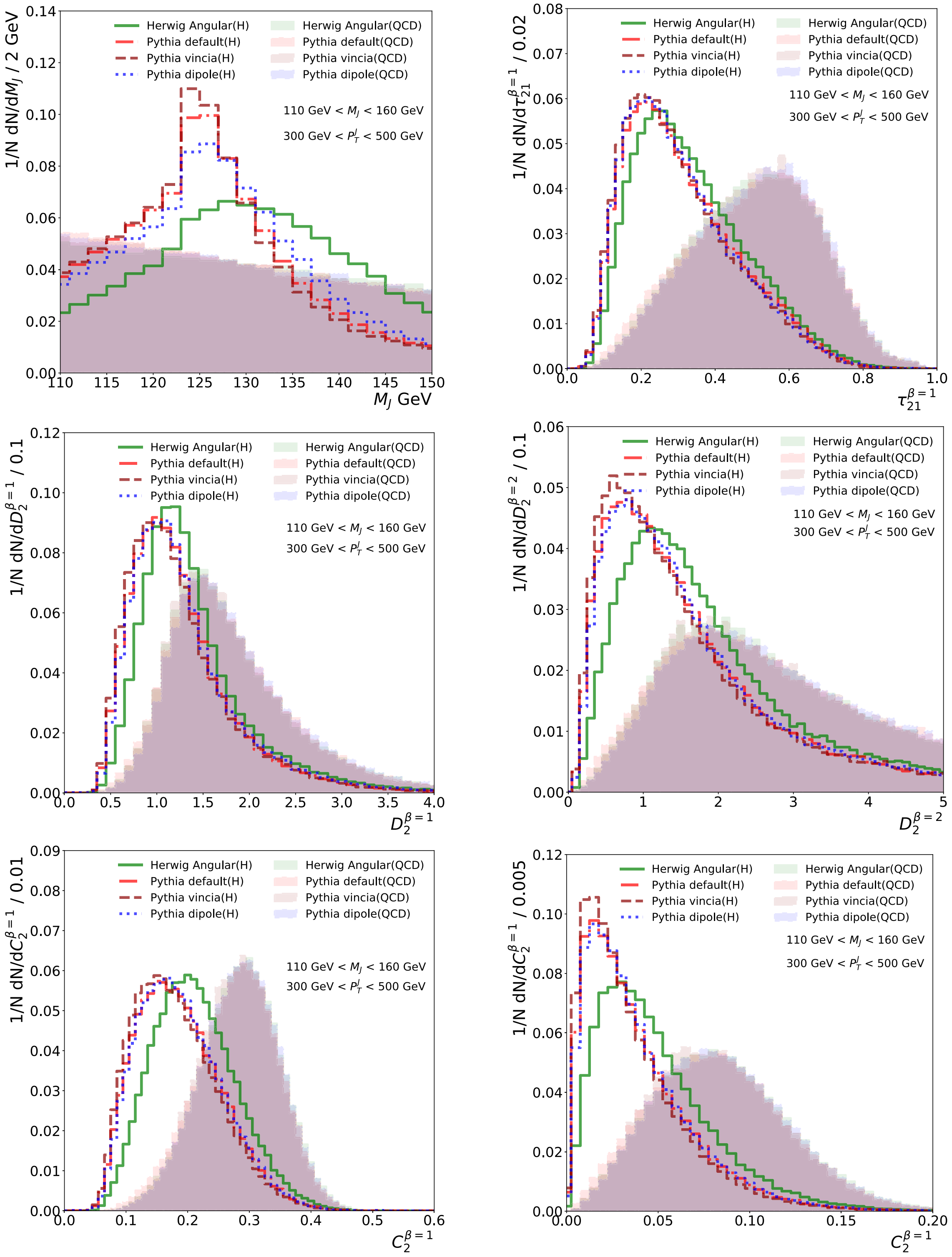
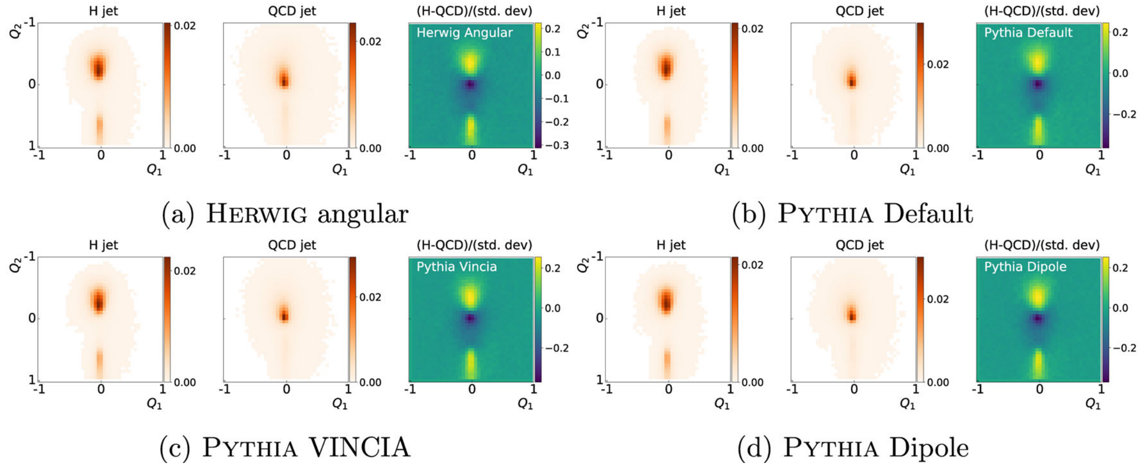


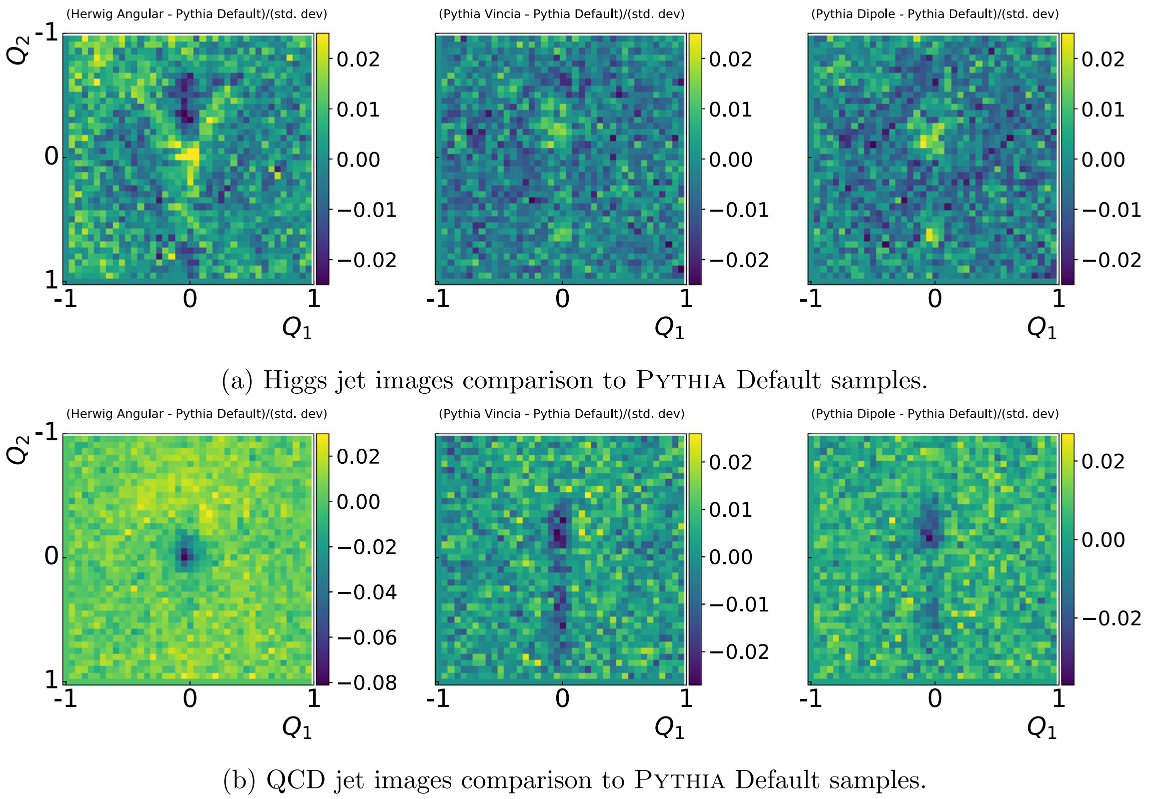
Fig. 1 The six high-level features used to distinguish boosted Higgs boson jets from QCD jets events



(a) HERWIG angular (b) PYTHIA Default

(c) PYTHIA VINCIA (d) PYTHIA Dipole

Fig. 2 Low-level features. The average of 40000 Higgs-like jet images in the charged p_T channel (left column and middle column). Q_1 and Q_2 denote the new axes after the jet's axis is centralized and rotated. The intensity in each pixel is the sum of the charged particle p_T. The total intensity in each image is normalized to unity. Images in right column are the average difference between Higgs jet and QCD jet images



(a) Higgs jet images comparison to PYTHIA Default samples.

(b) QCD jet images comparison to PYTHIA Default samples.

Fig. 3 The average difference between the other generators and the PYTHIA Default showering **a** Higgs-like jet images, and **b** QCD jet images. Q_1 and Q_2 denote the new axes after the jet's axis is centralized and rotated

Details of the CNN are as follows. The convolution filter is 5×5, the maximum pooling layers are 2×2, and the stride length is 1. ReLU activation functions are used for all intermediate layers of the neural network. The first convolution layer has 96 filters and the second convolution layer in each stream has 32 filters. A flatten layer is used after the second maximum pooling layer. Two dense layers are connected to the flatten layer with 350 and 400 neurons, respectively. Before the output layer, Dropout regularization is added with a dropout rate = 0.01. As for the dense network, the last activation is a sigmoid function and binary cross entropy is optimized during training. The AdaDelta optimizer [53]

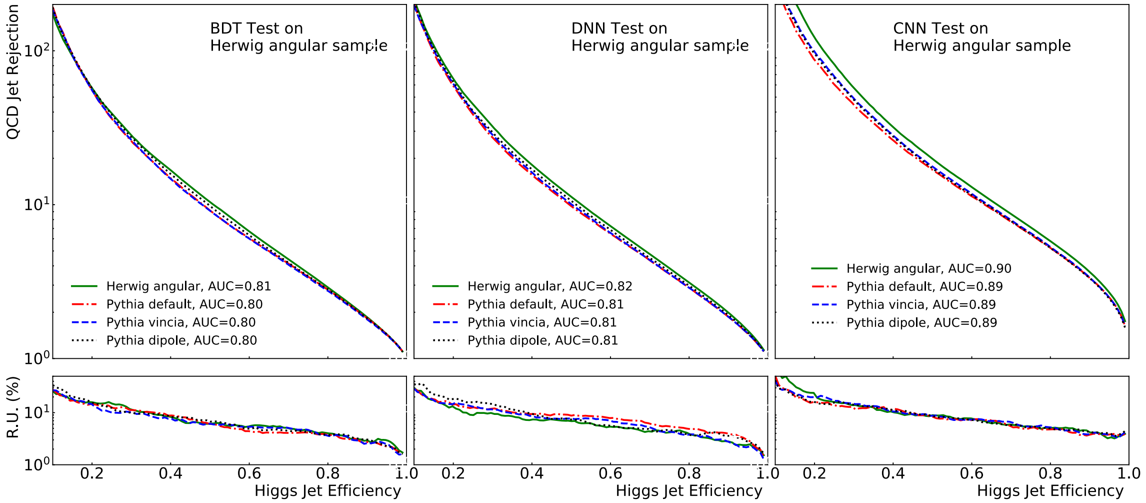


Fig. 4 The QCD rejection (inverse QCD efficiency) as a function of the Higgs jet efficiency for classifiers applied to HERWIG angular jet from four PSMC algorithms. The bottom panel shows the relative uncertainties

with learning rate 6.0216×10^{-3} is used to select the network weights. The KerasTuner [49] is used to get the best configuration of hyperparameters. The same setup as for the dense network is used to run the CNN.

4 Results

In this study, the receiver operating characteristic curve (ROC), the area under the ROC curve (AUC), the maximum significance improvement characteristic (SIC) and rejection (inverse background efficiency) at 50% signal efficiency are used to be metrics to quantify the universality. The AUC is between 0.5 (poor classification performance) and 1 (maximum classification performance). The SIC is the signal efficiency divided by the square root of the background efficiency and represents by how much (as a multiplicative factor) the significance would improve with a given threshold on the classifier score. The maximum SIC is simply the maximum SIC attained across all thresholds. In order to quantify the variation from classifier training itself, the performance is evaluated by k-fold cross-validation technique with $k = 50$. In this procedure, the datasets are randomly partitioned into 50 parts and for each one, the other 49 sets are used for constructing the classifier. The mean and spread over the folds is used to quantify the model performance.

Figure 4 shows four classifiers trained on various simulations and then tested on the same HERWIG dataset. Overall, the CNN has the best performance and the DNN is marginally better than the BDT. The DNN and BDT are trained on the same features and given the relatively low-dimensionality of the problem, it is unsurprising that the two models have a similar performance. Overall, the performance is nearly identical for all training sets. This is even true for the CNN, which has access to low-level substructure information inside the jets. The insensitivity to the training set is in stark contrast to the sensitivity of the test set, as summarized in detail below. Additional results can be found in Appendix A.

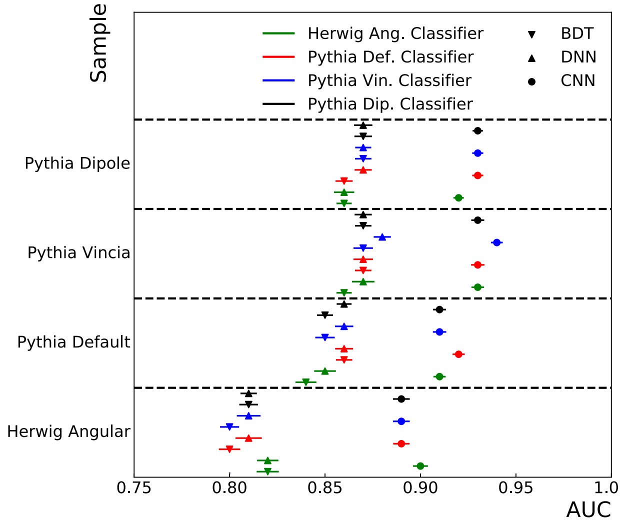


Fig. 5 The performance of classifiers as quantified by the AUC when training on a given PSMC (color) and testing on the PSMC specified on the vertical axis. The symbols represent the type of model (BDT, DNN, CNN). The error bars represent the standard deviation over the k folds

The performance of Fig. 4 for all combinations of train and test sets for the three machine learning models are summarized in Figs. 5, 6 and 7. Starting with Fig. 5, we observe that there is a significant spread in performance across test sets (rows). The difference between Higgs jets and QCD jets is smaller for HERWIG compared with PYTHIA by almost 10%. However, the spread in performance for a given test set is about 1%. Similar trends are present for the rejection at a fixed efficiency (Fig. 6) and maximum SIC (Fig. 7) plots, albeit with larger sensitivities to the machine learning training.

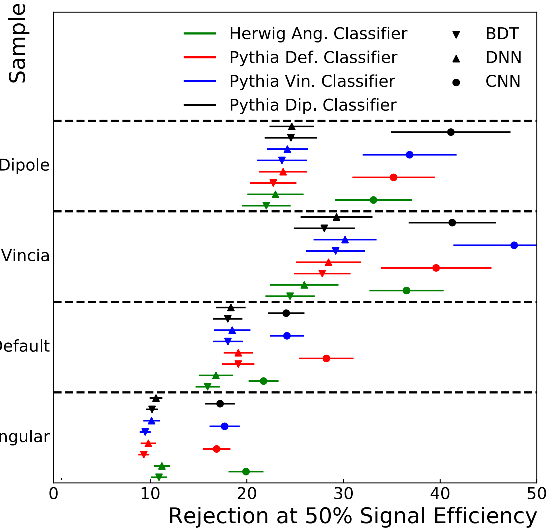


Fig. 6 The performance of classifiers as quantified by the rejection at a fixed signal efficiency of 50% when training on a given PSMC (color) and testing on the PSMC specified on the vertical axis. The symbols represent the type of model (BDT, DNN, CNN). The error bars represent the standard deviation over the *k* folds

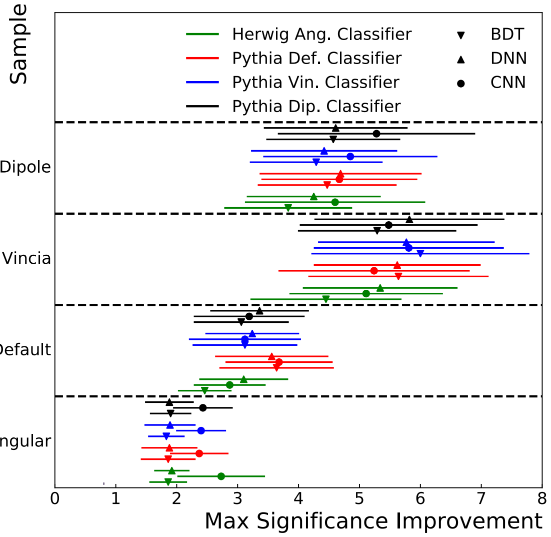


Fig. 7 The performance of classifiers as quantified by the maximum significance improvement when training on a given PSMC (color) and testing on the PSMC specified on the vertical axis. The symbols represent the type of model (BDT, DNN, CNN). The error bars represent the standard deviation over the *k* folds

5 Conclusions and outlook

We have explored the universality of classifiers trained on hadronic jet tagging. In particular, we have studied the sensitivity of the learned classifier to the Parton Shower Monte Carlo program used during training. While the modeling of the hadronic structure differs significantly among PSMCs, we find that the actual function learned is nearly independent of the training set. This gives us confidence that a classifier trained on one PSMC and tested on another (or data) will likely still be optimal. Although it is not directly a source of uncertainty for physics analysis, this observation has important implications for making the best use of our data. The classifier universality does not mean that the systematic uncertainty from hadronic modeling is small as bias and optimality are separate concepts (see e.g., Ref. [8]).

The universality not only has important experimental implications, but also motivates further theoretical studies. As in the quark versus gluon jet example referenced in Sect. 1, the universality of the classifiers suggests that a theoretical explanation of the classification performance may be attainable as it should be insensitive to the detailed modeling assumptions of a particular PSMC program. We look forward to studies in this direction.

Uncertainty quantification is a critical component of any analysis at the LHC and this task is particularly challenging for analysis strategies like machine learning that are sensitive to low-level hadronic modeling. While determining systematic uncertainties on the potential bias of a result from hadronic modeling is still an active area of research and development, we have shown that at least the optimality of machine learning classifiers is relatively insensitive to hadronic modeling. While we have observed this disconnect between bias and optimality for Higgs jet tagging, we conjecture that this is a generic feature of QCD and it may also be present in other systems at the LHC and beyond.

Acknowledgements We thank J. Thaler for useful feedback on the manuscript. K. Cheung and Y.-L.Chung are supported by the Taiwan MoST with the grant number MOST-110-2112-M-007-017-MY3. S.-C. Hsu is supported by the U.S. Department of Energy, Office of Science, Office of Early Career Research Program under Award number DE-SC0015971. B. Nachman is supported by the U.S. Department of Energy, Office of Science under contract DE-AC02-05CH11231.

Data Availability Statement This manuscript has associated data in a data repository. [Authors' comment: The data are available upon reasonable request.]

Funded by SCOAP^3. SCOAP^3 supports the goals of the International Year of Basic Sciences for Sustainable Development.

Appendix A: Remaining results

See Tables 1, 2, 3, 4, 5, 6, 7, Figs. 8, 9 and 10.

Table 1 Area under the curve, rejection at 50% signal efficiency and maximum significance improvement when testing on HERWIG for each trained classifier. The last rows are the average and standard deviation over the mean values from the other rows

Varied trained classifiers, test on HERWIG angular sample			
Trained model	Classifier type		
	BDT	Dense neural network	CNN
Metric: area under the curve			
Herwig angular	0.8193 ± 0.0058	0.8219 ± 0.0048	0.8991 ± 0.0039
Pythia default	0.8031 ± 0.0056	0.8079 ± 0.0080	0.8878 ± 0.0043
Pythia vincia	0.8043 ± 0.0050	0.8090 ± 0.0058	0.8897 ± 0.0044
Pythia dipole	0.8096 ± 0.0049	0.8141 ± 0.0051	0.8878 ± 0.0044
Average ± Std.	0.8091 ± 0.0064	0.8132 ± 0.0055	0.8911 ± 0.0047
Metric: rejection at 50% signal efficiency			
Herwig angular	10.91 ± 0.84	11.21 ± 0.83	19.91 ± 1.81
Pythia default	9.34 ± 0.57	9.81 ± 0.80	16.87 ± 1.43
Pythia vincia	9.48 ± 0.57	10.14 ± 0.85	17.70 ± 1.56
Pythia dipole	10.19 ± 0.64	10.60 ± 0.66	17.23 ± 1.55
Average ± Std.	9.98 ± 0.63	10.44 ± 0.52	17.93 ± 1.18
Metric: max significance improvement			
Herwig angular	1.86 ± 0.31	1.92 ± 0.29	2.73 ± 0.72
Pythia default	1.86 ± 0.45	1.88 ± 0.46	2.37 ± 0.48
Pythia cincia	1.83 ± 0.30	1.89 ± 0.42	2.40 ± 0.41
Pythia dipole	1.90 ± 0.34	1.88 ± 0.40	2.43 ± 0.49
Average ± Std.	1.87 ± 0.03	1.89 ± 0.02	2.48 ± 0.14

Table 2 Area under the curve, rejection at 50% signal efficiency and maximum significance improvement when testing on PYTHIA default for each trained classifier. The last rows are the average and standard deviation over the mean values from the other rows

Varied trained classifiers, test on PYTHIA default sample			
Trained model	Classifier ttype		
	BDT	Dense neural network	CNN
Metric: area under the curve			
Herwig angular	0.8439 ± 0.0055	0.8476 ± 0.0056	0.9064 ± 0.0032
Pythia default	0.8582 ± 0.0043	0.8590 ± 0.0043	0.9174 ± 0.0032
Pythia vincia	0.8545 ± 0.0051	0.8564 ± 0.0042	0.9103 ± 0.0035
Pythia dipole	0.8541 ± 0.0042	0.8561 ± 0.0043	0.9090 ± 0.0034
Average ± Std.	0.8527 ± 0.0053	0.8548 ± 0.0043	0.9107 ± 0.0041
Metric: rejection at 50% signal efficiency			
Herwig angular	15.94 ± 1.25	16.80 ± 1.78	21.73 ± 1.55
Pythia default	19.11 ± 1.67	19.11 ± 1.51	28.23 ± 2.81
Pythia vincia	18.04 ± 1.57	18.48 ± 1.89	24.15 ± 1.75
Pythia dipole	18.03 ± 1.51	18.35 ± 1.52	24.07 ± 1.89
Average ± Std.	17.78 ± 1.15	18.18 ± 0.85	24.55 ± 2.34
Metric: max significance improvement			
Herwig angular	2.46 ± 0.44	3.10 ± 0.73	2.87 ± 0.59
Pythia default	3.64 ± 0.94	3.56 ± 0.93	3.68 ± 0.88
Pythia vincia	3.12 ± 0.86	3.24 ± 0.77	3.12 ± 0.92
Pythia dipole	3.06 ± 0.78	3.36 ± 0.81	3.19 ± 0.91
Average ± Std.	3.07 ± 0.42	3.31 ± 0.17	3.21 ± 0.29

Table 3 Area under the curve, rejection at 50% signal efficiency and maximum significance improvement when testing on Pythia VINCIA for each trained classifier. The last rows are the average and standard deviation over the mean values from the other rows

Varied trained classifiers, test on Pythia VINCIA sample			
Trained model	Classifier type		
	BDT	Dense neural network	CNN
Metric: area under the curve			
Herwig angular	0.8625 ± 0.0040	0.8654 ± 0.0053	0.9259 ± 0.0033
Pythia default	0.8719 ± 0.0043	0.8736 ± 0.0046	0.9279 ± 0.0035
Pythia vincia	0.8748 ± 0.0051	0.8758 ± 0.0055	0.9351 ± 0.0031
Pythia dipole	0.8722 ± 0.0043	0.8739 ± 0.0048	0.9284 ± 0.0034
Average ± Std.	0.8704 ± 0.0047	0.8722 ± 0.0040	0.9293 ± 0.0035
Metric: rejection at 50% signal efficiency			
Herwig angular	24.47 ± 2.55	25.94 ± 3.54	36.52 ± 3.84
Pythia default	27.80 ± 2.93	28.45 ± 3.35	39.58 ± 5.73
Pythia vincia	29.19 ± 3.05	30.16 ± 3.26	47.66 ± 6.29
Pythia dipole	28.01 ± 3.16	29.28 ± 3.72	41.25 ± 4.51
Average ± Std.	27.37 ± 1.76	28.46 ± 1.57	41.25 ± 4.07
Metric: max significance improvement			
Herwig angular	4.45 ± 1.24	5.34 ± 1.27	5.11 ± 1.26
Pythia default	5.64 ± 1.48	5.62 ± 1.37	5.24 ± 1.57
Pythia vincia	6.00 ± 1.79	5.77 ± 1.45	5.81 ± 1.56
Pythia dipole	5.29 ± 1.30	5.82 ± 1.56	5.48 ± 1.46
Average ± Std.	5.34 ± 0.58	5.64 ± 0.19	5.41 ± 0.27

Table 4 Area under the curve, rejection at 50% signal efficiency and maximum significance improvement when testing on Pythia dipole for each trained classifier. The last rows are the average and standard deviation over the mean values from the other rows

Varied trained classifiers, test on Pythia dipole sample			
Trained model	Classifier type		
	BDT	Dense neural network	CNN
Metric: area under the curve			
Herwig angular	0.8601 ± 0.0040	0.8628 ± 0.0049	0.9227 ± 0.0027
Pythia default	0.8647 ± 0.0045	0.8671 ± 0.0047	0.9246 ± 0.0029
Pythia vincia	0.8654 ± 0.0043	0.8678 ± 0.0050	0.9265 ± 0.0029
Pythia dipole	0.8681 ± 0.0046	0.8694 ± 0.0054	0.9308 ± 0.0027
Average ± Std.	0.8646 ± 0.0029	0.8668 ± 0.0024	0.9261 ± 0.0030
Metric: rejection at 50% signal efficiency			
Herwig angular	22.01 ± 2.52	22.96 ± 2.91	33.10 ± 3.97
Pythia default	22.73 ± 2.40	23.75 ± 2.48	35.18 ± 4.26
Pythia vincia	23.63 ± 2.58	24.19 ± 2.12	36.84 ± 4.86
Pythia dipole	24.56 ± 2.72	24.65 ± 2.31	41.11 ± 6.17
Average ± Std.	23.23 ± 0.96	23.89 ± 0.62	36.56 ± 2.94
Metric: max significance improvement			
Herwig angular	3.83 ± 1.05	4.25 ± 1.10	4.60 ± 1.48
Pythia default	4.47 ± 1.14	4.69 ± 1.33	4.67 ± 1.28
Pythia vincia	4.29 ± 1.09	4.42 ± 1.20	4.85 ± 1.43
Pythia dipole	4.57 ± 1.10	4.61 ± 1.18	5.28 ± 1.62
Average ± Std.	4.29 ± 0.28	4.49 ± 0.17	4.85 ± 0.26

Table 5 Area under the curve, rejection at 50% signal efficiency and maximum significance improvement for the BDT model. The last rows are the average and standard deviation over the mean values from the other rows

Varied trained BDT models test on fixed sample				
Trained model	Fixed sample			
	Herwig angular	Pythia default	Pythia vincia	Pythia dipole
Metric: area under the curve				
Herwig angular	0.8193 ± 0.0058	0.8439 ± 0.0055	0.8625 ± 0.0040	0.8601 ± 0.0040
Pythia default	0.8031 ± 0.0056	0.8582 ± 0.0043	0.8719 ± 0.0043	0.8647 ± 0.0045
Pythia vincia	0.8043 ± 0.0050	0.8545 ± 0.0051	0.8748 ± 0.0051	0.8654 ± 0.0043
Pythia dipole	0.8096 ± 0.0049	0.8541 ± 0.0042	0.8722 ± 0.0043	0.8681 ± 0.0046
Average ± Std.	0.8091 ± 0.0064	0.8527 ± 0.0053	0.8704 ± 0.0047	0.8646 ± 0.0029
Metric: rejection at 50% signal efficiency				
Herwig angular	10.91 ± 0.84	15.94 ± 1.25	24.47 ± 2.55	22.01 ± 2.52
Pythia default	9.34 ± 0.57	19.11 ± 1.67	27.80 ± 2.93	22.73 ± 2.40
Pythia vincia	9.48 ± 0.57	18.04 ± 1.57	29.19 ± 3.05	23.63 ± 2.58
Pythia dipole	10.19 ± 0.64	18.03 ± 1.51	28.01 ± 3.16	24.56 ± 2.72
Average ± Std.	9.98 ± 0.63	17.78 ± 1.15	27.37 ± 1.76	23.23 ± 0.96
Metric: max significance improvement				
Herwig angular	1.86 ± 0.31	2.46 ± 0.44	4.45 ± 1.24	3.83 ± 1.05
Pythia default	1.86 ± 0.45	3.64 ± 0.94	5.64 ± 1.48	4.47 ± 1.14
Pythia vincia	1.83 ± 0.30	3.12 ± 0.86	6.00 ± 1.79	4.29 ± 1.09
Pythia dipole	1.90 ± 0.34	3.06 ± 0.78	5.29 ± 1.30	4.57 ± 1.10
Average ± Std.	1.87 ± 0.03	3.07 ± 0.42	5.34 ± 0.58	4.29 ± 0.28

Table 6 Area under the curve, rejection at 50% signal efficiency and maximum significance improvement for the DNN model. The last rows are the average and standard deviation over the mean values from the other rows

Varied trained DNN models test on fixed sample				
Trained model	Fixed sample			
	Herwig angular	Pythia default	Pythia vincia	Pythia dipole
Metric: area under the curve				
Herwig angular	0.8219 ± 0.0048	0.8476 ± 0.0056	0.8654 ± 0.0053	0.8628 ± 0.0049
Pythia default	0.8079 ± 0.0080	0.8590 ± 0.0043	0.8736 ± 0.0046	0.8671 ± 0.0047
Pythia vincia	0.8090 ± 0.0058	0.8564 ± 0.0042	0.8758 ± 0.0055	0.8678 ± 0.0050
Pythia dipole	0.8141 ± 0.0051	0.8561 ± 0.0043	0.8739 ± 0.0048	0.8694 ± 0.0054
Average ± Std.	0.8132 ± 0.0055	0.8548 ± 0.0043	0.8722 ± 0.0040	0.8668 ± 0.0024
Metric: rejection at 50% signal efficiency				
Herwig angular	11.21 ± 0.83	16.80 ± 1.78	25.94 ± 3.54	22.96 ± 2.91
Pythia default	9.81 ± 0.80	19.11 ± 1.51	28.45 ± 3.35	23.75 ± 2.48
Pythia vincia	10.14 ± 0.85	18.48 ± 1.89	30.16 ± 3.26	24.19 ± 2.12
Pythia dipole	10.60 ± 0.66	18.35 ± 1.52	29.28 ± 3.72	24.65 ± 2.31
Average ± Std.	10.44 ± 0.52	18.18 ± 0.85	28.46 ± 1.57	23.89 ± 0.62
Metric: max significance improvement				
Herwig angular	1.92 ± 0.29	3.10 ± 0.73	5.34 ± 1.27	4.25 ± 1.10
Pythia default	1.88 ± 0.46	3.56 ± 0.93	5.62 ± 1.37	4.69 ± 1.33
Pythia vincia	1.89 ± 0.42	3.24 ± 0.77	5.77 ± 1.45	4.42 ± 1.20
Pythia dipole	1.88 ± 0.40	3.36 ± 0.81	5.82 ± 1.56	4.61 ± 1.18
Average ± Std.	1.89 ± 0.02	3.31 ± 0.17	5.64 ± 0.19	4.49 ± 0.17

Table 7 Area under the curve, rejection at 50% signal efficiency and maximum significance improvement for the CNN model. The last rows are the average and standard deviation over the mean values from the other rows

Varied trained CNN models test on fixed sample				
Trained model	Fixed sample			
	Herwig angular	Pythia default	Pythia vincia	Pythia dipole
Metric: area under the curve				
Herwig angular	0.8991 ± 0.0039	0.9064 ± 0.0032	0.9259 ± 0.0033	0.9227 ± 0.0027
Pythia default	0.8878 ± 0.0043	0.9174 ± 0.0032	0.9279 ± 0.0035	0.9246 ± 0.0029
Pythia vincia	0.8897 ± 0.0044	0.9103 ± 0.0035	0.9351 ± 0.0031	0.9265 ± 0.0029
Pythia dipole	0.8878 ± 0.0044	0.9090 ± 0.0034	0.9284 ± 0.0034	0.9308 ± 0.0027
Average ± Std.	0.8911 ± 0.0047	0.9107 ± 0.0041	0.9293 ± 0.0035	0.9261 ± 0.0030
Metric: rejection at 50% signal efficiency				
Herwig angular	19.91 ± 1.81	21.73 ± 1.55	36.52 ± 3.84	33.10 ± 3.97
Pythia default	16.87 ± 1.43	28.23 ± 2.81	39.58 ± 5.73	35.18 ± 4.26
Pythia vincia	17.70 ± 1.56	24.15 ± 1.75	47.66 ± 6.29	36.84 ± 4.86
Pythia dipole	17.23 ± 1.55	24.07 ± 1.89	41.25 ± 4.51	41.11 ± 6.17
Average ± Std.	17.93 ± 1.18	24.55 ± 2.34	41.25 ± 4.07	36.56 ± 2.94
Metric: max significance improvement				
Herwig angular	2.73 ± 0.72	2.87 ± 0.59	5.11 ± 1.26	4.60 ± 1.48
Pythia default	2.37 ± 0.48	3.68 ± 0.88	5.24 ± 1.57	4.67 ± 1.28
Pythia vincia	2.40 ± 0.41	3.12 ± 0.92	5.81 ± 1.56	4.85 ± 1.43
Pythia dipole	2.43 ± 0.49	3.19 ± 0.91	5.48 ± 1.46	5.28 ± 1.62
Average ± Std.	2.48 ± 0.14	3.21 ± 0.29	5.41 ± 0.27	4.85 ± 0.26

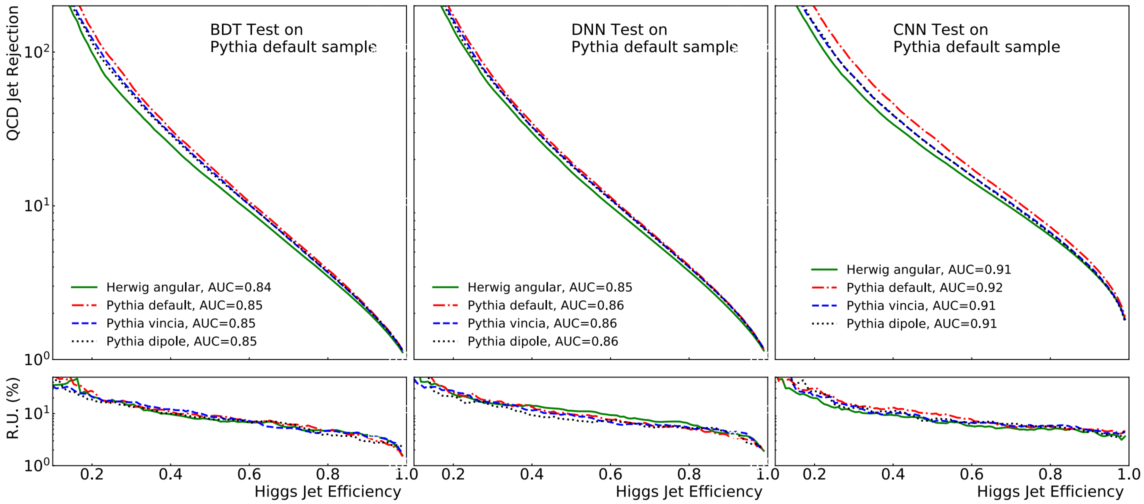


Fig. 8 The QCD rejection (inverse QCD efficiency) as a function of the Higgs jet efficiency for classifiers applied to PYTHIA default sample from four PSMC algorithms. The bottom panel shows the relative uncertainties

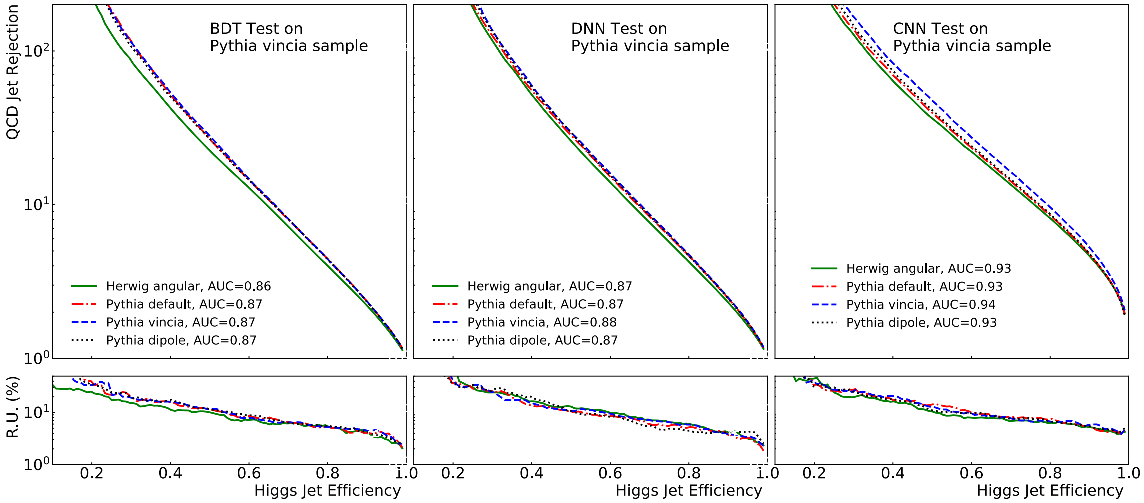


Fig. 9 The QCD rejection (inverse QCD efficiency) as a function of the Higgs jet efficiency for classifiers applied to Pythia VNICIA jet from four PSMC algorithms. The bottom panel shows the relative uncertainties

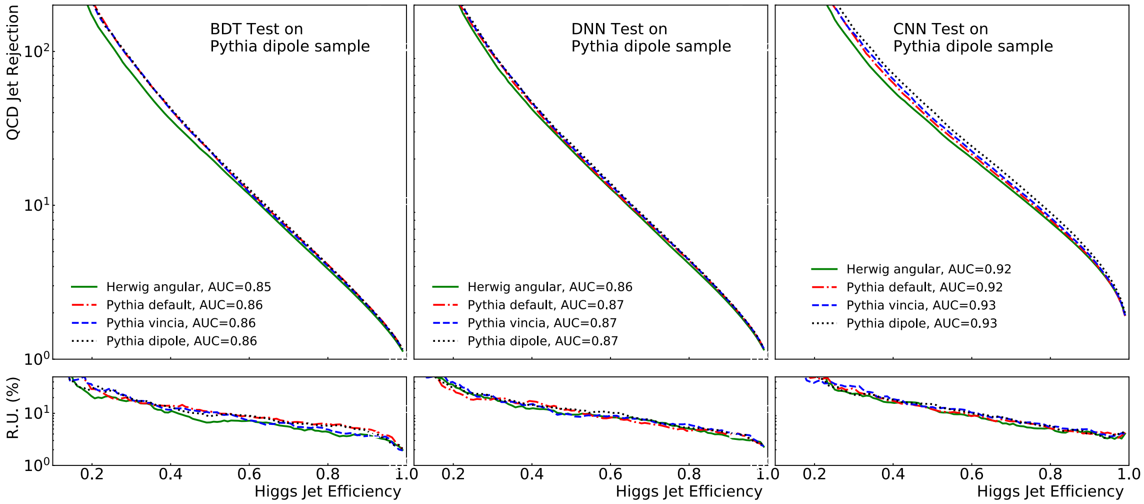


Fig. 10 The QCD rejection (inverse QCD efficiency) as a function of the Higgs jet efficiency for classifiers applied to Pythia dipole jet from four PSMC algorithms. The bottom panel shows the relative uncertainties

References

1. M. Feickert, B. Nachman, A living review of machine learning for particle physics. arXiv:2102.02770
2. A.J. Larkoski, I. Moult, B. Nachman, Jet substructure at the large hadron collider: a review of recent advances in theory and machine learning. Phys. Rep. **841**, 1–63 (2020). arXiv:1709.04464
3. D. Guest, K. Cranmer, D. Whiteson, Deep learning and its application to LHC physics. Ann. Rev. Nucl. Part. Sci. **68**, 161–181 (2018). arXiv:1806.11484
4. A. Radovic, M. Williams, D. Rousseau, M. Kagan, D. Bonacorsi, A. Himmel, A. Aurisano, K. Terao, T. Wongjirad, Machine learning at the energy and intensity frontiers of particle physics. Nature **560**(7716), 41–48 (2018)
5. D. Bourilkov, Machine and Deep learning applications in particle physics. Int. J. Mod. Phys. A **34**(35), 1930019 (2020). arXiv:1912.08245
6. G. Karagiorgi, G. Kasieczka, S. Kravitz, B. Nachman, D. Shih, Machine learning in the search for new fundamental physics. arXiv:2112.03769
7. A. Buckley et al., General-purpose event generators for LHC physics. Phys. Rep. **504**, 145–233 (2011). arXiv:1101.2599
8. B. Nachman, A guide for deploying deep learning in LHC searches: how to achieve optimality and account for uncertainty. SciPost Phys. **8**, 090 (2020). arXiv:1909.03081
9. G. Louppe, M. Kagan, K. Cranmer, Learning to pivot with adversarial networks, in *Advances in Neural Information Processing Systems*, vol. 30, ed. by I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Curran Associates, Inc., 2017). arXiv:1611.01046
10. C. Englert, P. Galler, P. Harris, M. Spannowsky, Machine learning uncertainties with adversarial neural networks. Eur. Phys. J. C **79**(1), 4 (2019). arXiv:1807.08763
11. S. Wunsch, S. Jórger, R. Wolf, G. Quast, Reducing the dependence of the neural network function to systematic uncertainties in the input space. arXiv:1907.11674

12. J.M. Clavijo, P. Glaysher, J.M. Katzy, Adversarial domain adaptation to reduce sample bias of a high energy physics classifier. arXiv:2005.00568
13. P. De Castro, T. Dorigo, INFERNO: inference-aware neural optimisation. Comput. Phys. Commun. **244**, 170–179 (2019). arXiv:1806.04743
14. A. Ghosh, B. Nachman, D. Whiteson, Uncertainty-aware machine learning for high energy physics. Phys. Rev. D **104**(5), 056026 (2021). arXiv:2105.08742
15. N. Simpson, L. Heinrich, neos: End-to-end-optimised summary statistics for high energy physics, in *20th International Workshop on Advanced Computing and Analysis Techniques in Physics Research: AI Decoded—Towards Sustainable, Diverse, Performant and Effective Scientific Computing, 3* (2022). arXiv:2203.05570
16. A. Ghosh, B. Nachman, A cautionary tale of decorrelating theory uncertainties. arXiv:2109.08159
17. B. Nachman, C. Shimmin, AI safety for high energy physics. arXiv:1910.08606
18. A. Stein, X. Coubez, S. Mondal, A. Novak, A. Schmidt, Improving robustness of jet tagging algorithms with adversarial training. arXiv:2203.13890
19. P.T. Komiske, E.M. Metodiev, M.D. Schwartz, Deep learning in color: towards automated quark/gluon jet discrimination. JHEP **01**, 110 (2017). arXiv:1612.01551
20. J. Bellm et al., Herwig 7.0/Herwig++ 3.0 release note. Eur. Phys. J. C **76**(4), 196 (2016). arXiv:1512.01178
21. T. Sjostrand, S. Mrenna, P.Z. Skands, A brief introduction to PYTHIA 8.1. Comput. Phys. Commun. **178**, 852–867 (2008). arXiv:0710.3820
22. J.A. Aguilar-Saavedra, Taming modeling uncertainties with mass unspecific supervised tagging. Eur. Phys. J. C **82**(3), 270 (2022). arXiv:2201.11143
23. C. Frye, A.J. Larkoski, J. Thaler, K. Zhou, Casimir meets Poisson: improved quark/gluon discrimination with counting observables. JHEP **09**, 083 (2017). arXiv:1704.06266
24. J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer, H.S. Shao, T. Stelzer, P. Torrielli, M. Zaro, The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations. JHEP **07**, 079 (2014). arXiv:1405.0301
25. J. Butterworth et al., PDF4LHC recommendations for LHC Run II. J. Phys. G **43**, 023001 (2016). arXiv:1510.03865
26. NNPDF Collaboration, R.D. Ball et al., Parton distributions for the LHC Run II. JHEP **04**, 040 (2015). arXiv:1410.8849
27. P. Skands, R. Verheyen, Multipole photon radiation in the Vincia parton shower. Phys. Lett. B **811**, 135878 (2020). arXiv:2002.04939
28. H. Brooks, C.T. Preuss, Efficient multi-jet merging with the Vincia sector shower. Comput. Phys. Commun. **264**, 107985 (2021). arXiv:2008.09468
29. N. Fischer, S. Prestel, M. Ritzmann, P. Skands, Vincia for hadron colliders. Eur. Phys. J. C **76**(11), 589 (2016). arXiv:1605.06142
30. T. Sjöstrand, Jet fragmentation of multiparton configurations in a string framework. Nucl. Phys. B **248**(2), 469–502 (1984)
31. B. Andersson, G. Gustafson, G. Ingelman, T. Sjöstrand, Parton fragmentation and string dynamics. Phys. Rep. **97**(2), 31–145 (1983)
32. B. Webber, A qcd model for jet fragmentation including soft gluon interference. Nucl. Phys. B **238**(3), 492–528 (1984)
33. J.-C. Winter, F. Krauss, G. Soff, A modified cluster hadronization model. Eur. Phys. J. C **36**, 381–395 (2004). arXiv:hep-ph/0311085
34. N. Dawe, E. Rodrigues, H. Schreiner, B. Ostdiek, D. Kalinkin, M.R.S. Meehan, aryan26roy, and domen13, scikit-hep/pyjet: Version 1.8.2, Jan (2021)
35. M. Cacciari, G.P. Salam, G. Soyez, yFastJet user manual. Eur. Phys. J. C **72**, 1896 (2012). arXiv:1111.6097
36. M. Cacciari, G.P. Salam, G. Soyez, The anti-k_t jet clustering algorithm. JHEP **04**, 063 (2008). arXiv:0802.1189
37. J. Lin, M. Freytsis, I. Moult, B. Nachman, Boosting $H \to b\bar{b}$ with Machine Learning. JHEP **10**, 101 (2018). arXiv:1807.10768
38. M. Cacciari, G.P. Salam, G. Soyez, The catchment area of jets. JHEP **04**, 005 (2008). arXiv:0802.1188
39. A. Buckley, C. Pollard, yQCD-aware partonic jet clustering for truth-jet flavour labelling. Eur. Phys. J. C **76**(2), 71 (2016). arXiv:1507.00508
40. J. Thaler, K. Van Tilburg, Identifying boosted objects with N-subjettiness. JHEP **03**, 015 (2011). arXiv:1011.2268
41. J. Thaler, K. Van Tilburg, Maximizing boosted top identification by minimizing N-subjettiness. JHEP **02**, 093 (2012). arXiv:1108.2701
42. A.J. Larkoski, I. Moult, D. Neill, Power counting to better jet observables. JHEP **12**, 009 (2014). arXiv:1409.6298
43. A.J. Larkoski, G.P. Salam, J. Thaler, Energy correlation functions for jet substructure. JHEP **06**, 108 (2013). arXiv:1305.0007
44. Particle Data Group, Review of particle physics. Progr. Theor. Exp. Phys. **2020**(08), 083C01 (2020)
45. J. Cogan, M. Kagan, E. Strauss, A. Schwarztman, Jet-images: computer vision inspired techniques for jet tagging. JHEP **02**, 118 (2015). arXiv:1407.5675
46. L. de Oliveira, M. Kagan, L. Mackey, B. Nachman, A. Schwartzman, Jet-images—deep learning edition. JHEP **07**, 069 (2016). arXiv:1511.05190
47. L. de Oliveira, M. Paganini, B. Nachman, Learning particle physics by example: location-aware generative adversarial networks for physics synthesis. arXiv:1701.05927
48. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
49. T. O'Malley, E. Bursztein, J. Long, ß. Chollet, H. Jin, L. Invernizzi, et al. KerasTuner. (2019). https://github.com/keras-team/keras-tuner
50. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(56), 1929–1958 (2014)
51. D.P. Kingma, J. Ba, Adam: a method for stochastic optimization (2014). https://doi.org/10.48550/ARXIV.1412.6980; arXiv:1412.6980
52. M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org
53. M.D. Zeiler, ADADELTA: an adaptive learning rate method, CoRR. (2012). arXiv:1212.5701