Title
A cautionary tale of decorrelating theory uncertainties

Permalink
https://escholarship.org/uc/item/9fh890hc

Journal
European Physical Journal C, 82(1)

ISSN
1434-6044

Authors
Ghosh, Aishik
Nachman, Benjamin

Publication Date
2022

DOI
10.1140/epjc/s10052-022-10012-w

Peer reviewed

Eur. Phys. J. C (2022) 82:46
https://doi.org/10.1140/epjc/s10052-022-10012-w

THE EUROPEAN
PHYSICAL JOURNAL C

Regular Article - Theoretical Physics

A cautionary tale of decorrelating theory uncertainties

Aishik Ghosh[1,2,a], **Benjamin Nachman**[2,3,b]

[1] Department of Physics and Astronomy, University of California, Irvine, CA 92697, USA
[2] Physics Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA
[3] Berkeley Institute for Data Science, University of California, Berkeley, CA 94720, USA

Received: 3 October 2021 / Accepted: 10 January 2022 / Published online: 18 January 2022

Abstract A variety of techniques have been proposed to train machine learning classifiers that are independent of a given feature. While this can be an essential technique for enabling background estimation, it may also be useful for reducing uncertainties. We carefully examine theory uncertainties, which typically do not have a statistical origin. We will provide explicit examples of two-point (fragmentation modeling) and continuous (higher-order corrections) uncertainties where decorrelating significantly reduces the apparent uncertainty while the true uncertainty is much larger. These results suggest that caution should be taken when using decorrelation for these types of uncertainties as long as we do not have a complete decomposition into statistically meaningful components.

Contents

1 Introduction

Modern machine learning classifiers hold great promise for increasing the sensitivity of high energy physics data analyses [1–8]. Typically, a classifier is trained using simulated data and then the number of events passing a fixed threshold on the classifier output in data and in simulation is counted. A comparison between these counts is then used to estimate model parameters such as masses, couplings, and new physics cross sections. Theoretical and experimental uncertainties on the final result are accounted for by varying an aspect of the simulation and recomputing the predicted count using the nominal classifier. The uncertainties from the simulation model used for training affect the optimality of the classifier itself [9], but typically do not cause a bias and can be accounted for [10] by using parameterized classifiers [11,12].

A variety of techniques have been proposed to render a classifier independent of a given feature [13–24]. This has become an essential tool for resonance searches, where thresholds on the classifier output must not sculpt bumps in a given spectrum so that the Standard Model background can be estimated using sideband fits. The same methodology has also been proposed to reduce the impact of systematic uncertainties on classifier-based inference [25–28]. If such a classifier does not depend on a particular nuisance parameter, then the count computed when the parameter is varied will be the same as the nominal value. This means that the uncertainty on the parameter(s) of interest will appear to be reduced.

In the case that the systematic uncertainty is decomposed into its most fundamental components, each with a clear statistical interpretation, the above would be the end of the story. The systematic uncertainty can be reduced through decorrelation and this would be useful if the classification performance does not rely strongly on the value of the nuisance parameters (otherwise, it may be better to profile instead [10]). However, theory uncertainties almost never satisfy these conditions. These uncertainties are the result of approximations when performing calculations and are also due to parameter freedom in phenomenological models that are needed when first-principles calculations are not possible. The canonical examples for these two types of uncertainties

[a] e-mail: aishikghosh@lbl.gov
[b] e-mail: bpnachman@lbl.gov (corresponding author)

are perturbative uncertainties from series truncation and fragmentation modeling. For the former, calculations are truncated at a fixed order in perturbation theory and the result depends on unphysical scales. These scales are varied typically by factors of two in order to determine the uncertainty. Since the scales can be varied continuously, we refer to these as 'continuous uncertainties'. Fragmentation modeling uncertainties are often evaluated by comparing two different models, such as the string model [29,30] in the PYTHIA [31,32] parton shower Monte Carlo (PSMC) and the cluster model [33,34] in the HERWIG [35,36] PSMC. These variations are then interpreted as a one standard deviation uncertainty and combined with other sources of uncertainty in a final statistical analysis. Since only two variations of fragmentation modeling are usually available, we refer to these as 'two-point uncertainties'.

Continuous and two-point variations are ad hoc techniques commonly used in the particle physics community to have some handle over these difficult-to-estimate uncertainties. Generating multiple simulations from different fragmentation models allows us to probe two points in an underexplored theory space of fragmentation models. The difference between these two models provides only a rough estimate of how different nature may be to either of them. Varying unphysical scales would not change the observed physics if the full calculations could be performed. The sensitivity of our simulations to these scale variations therefore provides a rough estimate of the uncertainty associated with truncating the calculations at lower order. While the numerical value of uncertainties coming from statistically interpretable origins is well trusted, the kind of theoretical uncertainties discussed above only provide a rough estimate. This is in contrast to experimental nuisance parameters (that give rise to experimental uncertainties), including the jet energy scale. Such nuisance parameters are constrained using calibration datasets. The statistical uncertainty of the control region becomes a systematic uncertainty for the experimental nuisance parameters. This justifies treating the corresponding nuisance parameters as (approximate) Gaussian random variables. A detailed discussion of the origin and validity of theory uncertainties is outside the scope of this paper.

We examine the interplay of decorrelation with theory uncertainties. In particular, we will show that constructing a classifier that is independent of a given theory nuisance parameter does not mean that the theory uncertainty is zero. Instead, it means that the only handle to determine the theory uncertainty is eliminated. Figure 1 illustrates the intuition behind why this might be the case. As concrete examples, we study fragmentation modeling in the context of classifying Lorentz-boosted W boson jet from QCD jets and factorization scale variations in the context of classifying t-channel single top quark events from W+jets events.

This paper is organized as follows. Section 2 briefly introduces existing decorrelation techniques. Numerical examples of both two-point and continuous uncertainties are provided in Sect. 3. The paper ends with conclusions and outlook in Sect. 4.

2 Decorrelation techniques

Let $x \in \mathbb{R}^n$ be the features used for classification. Suppose that there is a feature[1] $m \in \mathbb{R}$ that we want to be decorrelated from a classifier $f(x) : \mathbb{R}^n \to \mathbb{R}$. One can achieve this decorrelation by minimizing the following loss functional L:

$$L[f(x)] = \sum_{i\in S} L_{\text{class}}(f(x_i), 1) + \sum_{i\in B} w(m_i) L_{\text{class}}(f(x_i), 0) + \lambda \sum_{i\in B} L_{\text{decor}}(f(x_i), m_i), \tag{2.1}$$

where S and B represent signal and background events, respectively. The loss L_{class} is the classifier loss and is often the binary cross entropy loss $L_{\text{class}}(f(x), y) = y \log(f(x)) + (1-y)\log(1-f(x))$. The function $w(m)$ represents a weighting function and λ represents a hyperparameter that controls the strength of the decorrelation. Finally, L_{decor} is a term that penalizes any dependence between f and m. This last term in Eq. (2.1) is schematic as the decorrelation penalty often acts at the level of batches of events and not individual examples. Standard classification corresponds to $w(m) = 1$ and $\lambda = 0$. Decorrelation approaches include:

- Planing [37,38]: $\lambda = 0$ and $w(m_i) \approx p_S(m)/p_B(m)$ so that the marginal distribution of m is non-discriminatory after the reweighting.
- Adversaries [16,25,26,28]: $w(m) = 1, \lambda < 0$, and L_{decor} is the loss of a second neural network (adversary) that takes $f(x)$ as input and tries to learn some properties of m.
- Distance Correlation (DisCo) [19,23]: $w(m) = 1, \lambda > 0$, and the last term in Eq. (2.1) is the *distance correlation* [39–42] between $f(x)$ and m for the background.
- Flatness [21]: $w(m) = 1$, $\lambda > 0$, and $L_{\text{decor}} = \sum_m b_m \int |F_m(s) - F(s)|^2 \, \mathrm{d}s$ where the sum runs over mass bins, b_m is the fraction of candidates in bin m, F is the cumulative distribution function, and $s = f(x)$ is the classifier output. This is generalized to Moment Decorrelation (MoDE) in Ref. [24] to allow for a given dependence of f on m.

[1] This also applies to cases where m is multi-dimensional, but we restrict to the one-dimensional setting here for simplicity and because it is widely used.

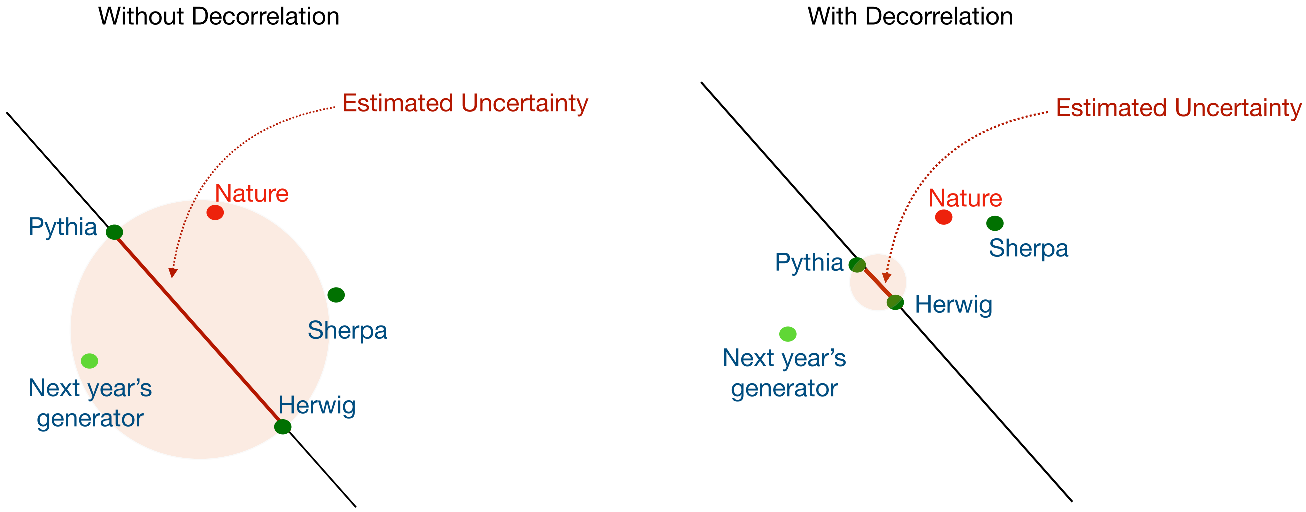

Fig. 1 An illustration of the potential impact of training a classifier to be decorrelated to two-point uncertainties. The distance between Pythia and Herwig is treated as the uncertainty. Left: Without decorrelation, the uncertainty covers nature even if nature does not lie on the line connecting Pythia and Herwig. Right: The distance between Pythia and Herwig is reduced due to the decorrelation requirement, resulting in a smaller estimate of the uncertainty, which no longer covers nature. These diagrams are meant only to be intuitive illustrations

In the examples below, we focus on the adversarial case as it is the most explored in the literature. However, the same ideas apply to all decorrelation methods.

3 Numerical examples

All neural networks are implemented using Keras [43] with the TensorFlow backend [44] and optimized with Adam [45].

3.1 Two-point uncertainty: fragmentation modeling

General purpose event generators use perturbation theory when they can and phenomenological models to describe non-perturbative effects such as hadronization. The standard procedure for estimating the uncertainty due to the model choice is to compare the predictions from two different models. This uncertainty is typically largest when the analysis strategy exploits subtle correlations in the high-dimensional radiation pattern. For example, tagging the origin of high p_T jets is a widely-studied scenario [46–48] for machine learning whereby the detailed jet substructure can be used for classification. In this section, we study Lorentz-boosted W boson tagging, where the signal is hadronically decaying, high p_T W bosons and the background is generic quark and gluon jets. A single large-radius jet is often sufficient to capture most of the W boson decay products and its two-prong substructure is distinct from typical quark and gluon jets.

Samples were generated with MadGraph5_aMC@NLO 2.7.3 [49] for modeling pp collisions at $\sqrt{s} = 13$ TeV. The `NNPDF23_nlo_as_0118` [50] parton distribution function is used. The hard-scattering events are passed to Pythia 8.303 [32] to simulate the parton shower and hadronization, using the default settings. Herwig 7.2.2 [35] with angularly-ordered showers and Sherpa 2.2.2 [51,52] with default settings are also used to model the parton shower and hadronization.[2] The jets are clustered by PyJet [53,54] and the anti-k_t [55] algorithm with radius parameter $R = 1.2$.

A set of high-level jet substructure features are used to distinguish W jets from QCD jets. These features are illustrated in Fig. 2 and briefly described in the following. The kinematics are probed with the jet mass and transverse momentum. Jet substructure observables include n-subjettiness ratio $\tau_{21} = \tau_2/\tau_1$ [56,57], and energy correlation function ratios $D_2^{(\beta)} = e_3^{(\beta)}/(e_2^{(\beta)})^3$ [58] and $C_2^{(\beta)} = e_3^{(\beta)}/(e_2^{(\beta)})^2$ [59], 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, we consider both $\beta = 1$ and $\beta = 2$.

As expected, the mass peaks near the W boson mass of 80 GeV [60] for the signal and has a broad distribution for the background. The signal peak is slightly higher than the W boson mass due to underlying event and other event con-

[2] While Herwig and Sherpa both use a cluster model for fragmentation, the actual Sherpa implementation is based on [34] and differs from Herwig in several respects.

tamination. This could be mitigated with grooming [61–65]. The jet p_T is not very discriminating by construction. The two-prong nature of the signal jets is quantified by low values of τ_{21}, D_2, and C_2.

A classifier is trained using the seven features presented in Fig. 2 to distinguish W jets from QCD jets. The nominal classifier is trained using the PYTHIA simulation and is parameterized as a neural network with two hidden layers of 50 nodes each. Rectified Linear Unit (ReLU) activations are used for the intermediate layers and the final output is passed through a sigmoid function. The binary cross entropy loss is used for training with a batch size of 100 and for 20 epochs. About 1 million events are used for each generator, with 50% for training and 50% for testing. None of these parameters were optimized, although minor variations were found to have little impact on performance. The performance of this nominal classifier evaluated on PYTHIA, HERWIG, and SHERPA is shown in Fig. 3. We focus on the region near 10–15% signal efficiency, which is a typical working point for LHC analyses. In this range, the background rejection (inverse QCD efficiency) is between a few hundred and a few thousand.

A second network is trained as part of an adversarial approach. This second network uses both PYTHIA and HERWIG events and minimizes the following loss:

$$L[f,g] = -\left(\sum_{i\in W}\log(f(x_i)) - \sum_{i\in \mathrm{QCD}}\log(1-f(x_i))\right) + \lambda\left(\sum_{i\in \mathrm{Pythia}}\log(g(f(x_i),y_i)) - \sum_{i\in \mathrm{Herwig}}\log(1-g(f(x_i),y_i))\right), \tag{3.1}$$

where f is the classifier, g is the adversary, $y_i = 0$ for W jets and $y_i = 1$ for QCD jets. Furthermore, $\lambda = 10$. Note that unlike Eq. (2.1), Eq. (3.1) has the labels as part of the function for the adversary. This means that the labels for the classifier are given as an input feature to the adversary, which allows the adversary to potentially learn separate decision functions for W jets and QCD jets. The classifier network f has the same composition as the nominal classifier described above: two hidden layers with 50 nodes each. The adversary has five hidden layers with 50 nodes each. As W jets are more different from QCD jets than PYTHIA jets are from HERWIG jets, the adversary has a more difficult task, which is why g has a more complex architecture. It was found that adding the label y_i to g as well as multiplying the gradient for the adversary by 10 improved performance and stability. The minimax nature of the optimization in Eq. (3.1) is implemented by connecting the adversary to the classifier via a gradient reversal layer [66] that multiplies the gradient by a fixed negative constant during backpropagation. The classifier network is then extracted after training for 20 epochs. When $\lambda = 0$, the performance was found to be the same as for the nominal case.[3]

Figure 3 shows that the performance of the adversarially trained classifier is worse than the nominal case. This drop in performance is the cost for building a classifier that is insensitive to fragmentation model variations. The difference between PYTHIA and HERWIG for the nominal classifier is about 40% at 10% W efficiency while it is only about 20% for the adversarially trained network.[4] The reduced difference may give the impression that the adversarially trained classifier has successfully learnt to be less sensitive to fragmentation model variations. However, the difference between SHERPA and PYTHIA is nearly the same for the nominal and the adversarially trained classifier. This means that the ‘true’ uncertainty would be significantly underestimated if only PYTHIA and HERWIG were available. It is often the case in an LHC analysis that only two fragmentation models are available. While the choice of SHERPA as the independent third generator is arbitrary, it is simply used in this study as a third point in the under-examined theory space of fragmentation modeling (Fig. 1), in order to demonstrate that the difference in performance of the classifier on an independent third point (whether another generator or nature) may not be well decorrelated. The result demonstrates the danger of training decorrelation methods on the same two generators that are then used to also estimate the theory uncertainty.

A curious reader may wonder why SHERPA does not lie within the range spanned between PYTHIA and HERWIG even for the nominal classifier (also alluded to in the illustration in Fig. 1). The uncertainties from these two generators of course do not restrict all other generators to lie within them, it treated as one standard deviation uncertainty[5] not as the maximum possible deviation. This study however reveals that applying decorrelation techniques would dramatically reduce the estimate of the uncertainty without necessarily reducing the differences to other generators or to nature.

[3] Note that when $\lambda = 0$, the adversarial setup is slightly different than the nominal configuration because both PYTHIA and HERWIG are used for training. This has little impact on the results – see Appendix A.

[4] It is possible this could be reduced with further hyperparameter tuning. We found some parameters that made this smaller, but with significant variation across trainings. The configuration reported here was found to be robust to retraining.

[5] To test the reliability of this uncertainty, one would need a large number of generators that span all possible ways of describing fragmentation modeling, and check how often they lie within the uncertainty bands; this is not possible in reality.

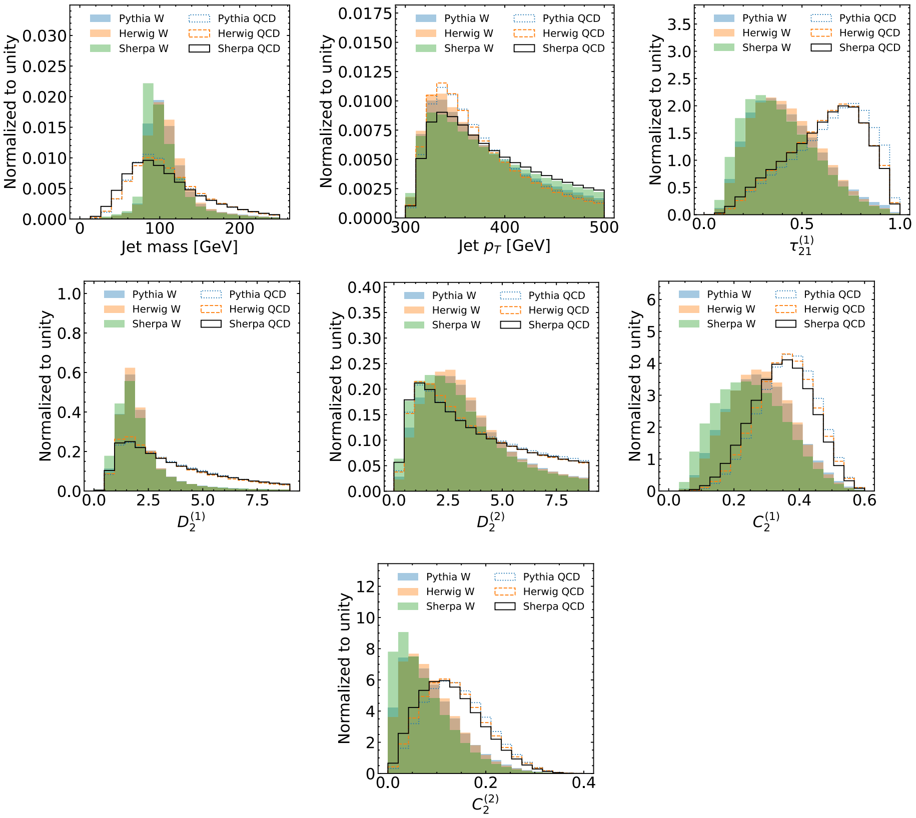

Fig. 2 The seven features used to train a classifier to distinguish boosted W boson jets from generic QCD jets events

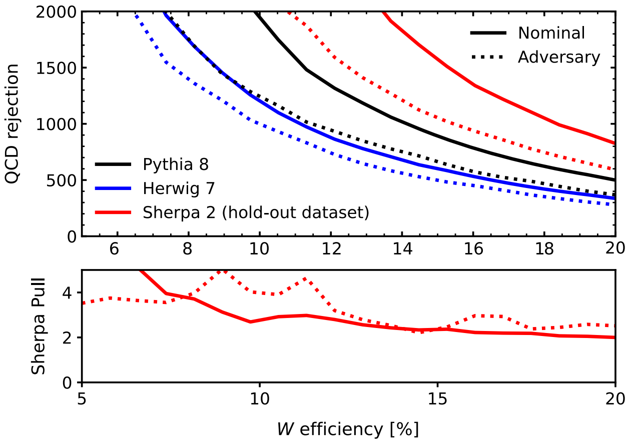

Fig. 3 The QCD rejection (inverse QCD efficiency) as a function of the W jet efficiency for classifiers applied to Pythia, Herwig, and Sherpa jets. The solid lines correspond to the nominal classifier trained with Pythia while the dotted lines correspond to the adversarial setup that uses both Pythia and Herwig (Sherpa is a hold-out dataset). The bottom panel shows the pull, which is the difference between Pythia and Sherpa divided by the uncertainty defined by the difference between Pythia and Herwig. While adversarial training reduces the difference in performance between Pythia and Herwig, the difference to Sherpa remains large, indicating that the true uncertainty will be underestimated if a third independent sample is unavailable

3.2 Continuous uncertainty: higher-order corrections

The uncertainty from truncating the order of a perturbative calculation is typically estimated by varying the unphysical scales. Usually, there are renormalization scale and factorization scale uncertainties. For simplicity, we focus here on the factorization scale, which dictates the separation between long- and short-distance physics. The standard procedure is to set the factorization scale to the typical momentum transfer in the problem.

To study the impact of factorization scale variations, we consider measurements of t-channel single top quark production. One of the main backgrounds for this process is W+jets production and machine learning is already used by ATLAS [67] and CMS [68] to enhance the signal. The semileptonic channel is studied as it has a much smaller background than the all-hadronic channel. The final state is characterized by an isolated lepton, missing transverse momentum, and jets.

Events are simulated using MadGraph5_aMC@NLO (MG5_aMC) 3.1.1 [49] interfaced with Pythia 8.244 [32] for the parton shower and Delphes 3.4.2 [69–71] for detector simulations with the default CMS card. Particle flow candi-

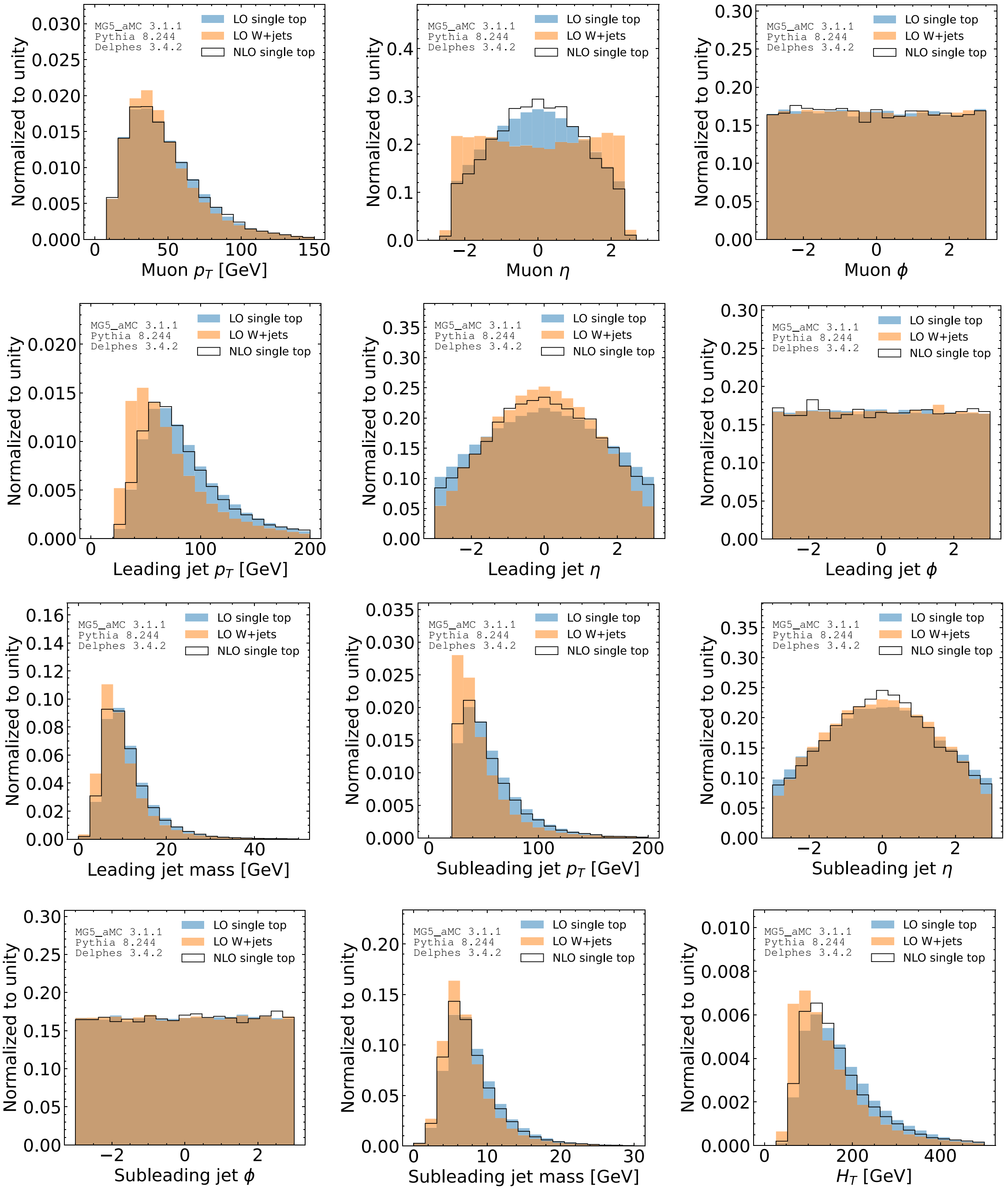

Fig. 4 The 12 features used to train a classifier to distinguish single top events from W+jets events

dates are used as inputs to jet clustering, implemented using FastJet 3.2.1 [54,72] and the anti-k_t algorithm [55] with radius parameter $R = 0.5$. For simplicity, W bosons are forced to decay into muons and events are required to have at least one isolated and identified muon using the default reconstruction algorithm in Delphes. Usually, one uses the highest precision method possible and then scale variations give the uncertainty from the finite truncation of the perturbative series. In order to compare with the 'true' uncertainty, we artificially truncate the series early and then use the higher-order calculation as the reference uncertainty. In particular, the nominal simulation is performed at leading order (LO) in the strong coupling constant and then an additional sample for the t-channel process is simulated at next-to-leading order (NLO).

For the machine learning, events are represented by 12 numbers: the three-momentum of the muon, the four-momentum of the leading two jets, and the scalar sum of

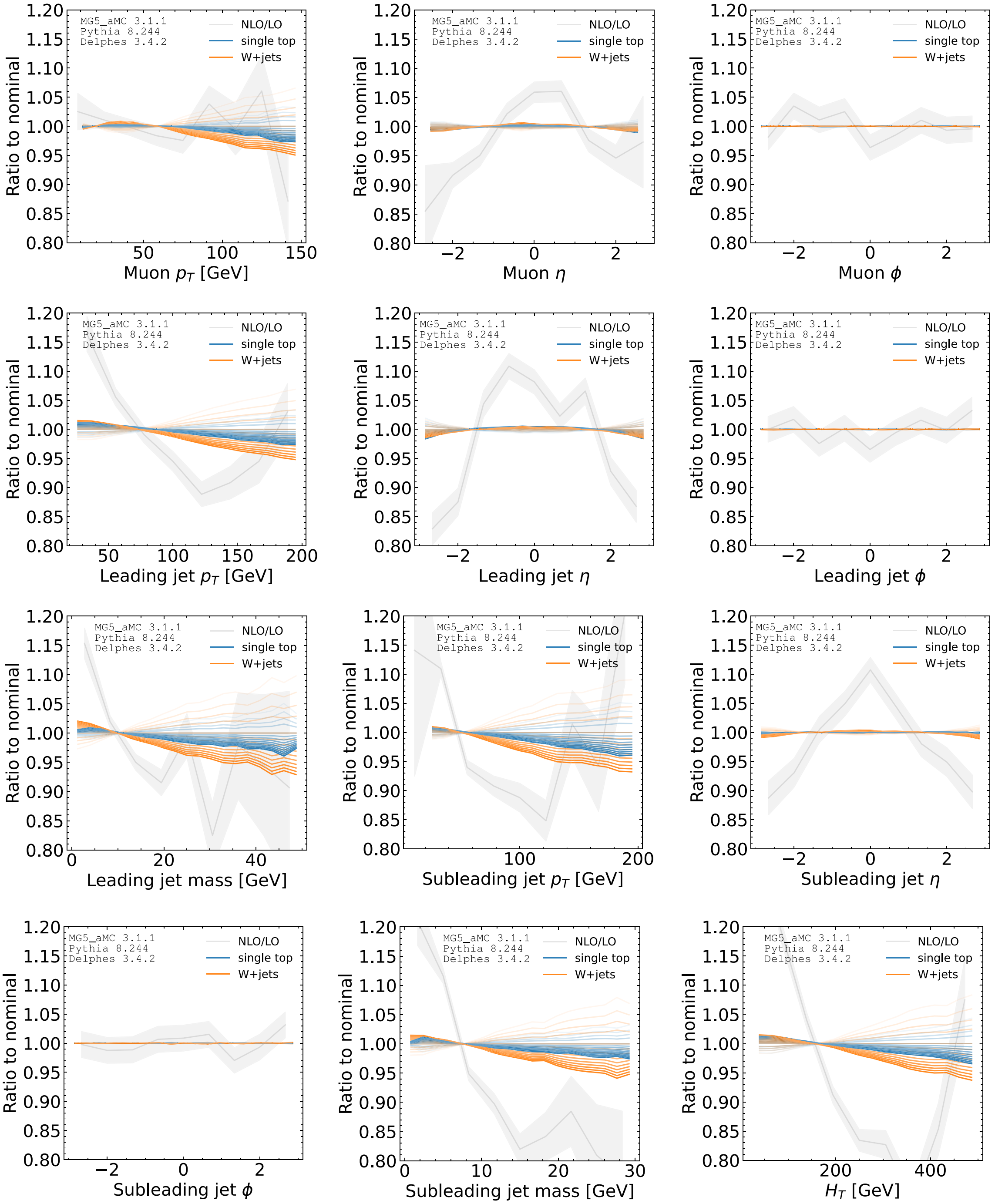

Fig. 5 The impact of factorization scale variations by a factor of 1/2 and 2, in increments of 0.1 (lighter colors are lower scales). In each case, histograms of the given observable with a particular factorization scale are normalized to unity and divided by the normalized, nominal histograms from Fig. 4. The single top NLO/LO differences are shown in grey, with the band representing the statistical uncertainty

the transverse momenta of all jets (H_T). Momenta are specified by p_T, η, and ϕ. Histograms for each of the observables for single top t-channel and W+jets are shown in Fig. 4. The jet p_T spectra are harder for single top compared with W jets and the muons (jets) tend to be more central (forward) for single top compared with W+jets.

The impact of factorization scale variations is shown in Fig. 5. All variations are normalized to unity, as the impact on the total cross section is not relevant for per-event classification performance. As expected, the variation for all ϕ observables is negligible and the biggest variation occurs for the transverse momenta.

The default performance for a classifier trained to distinguish single top events from W+jets events is shown in the top plot of Fig. 6. The W+rejection at a single top efficiency of 10% is about 75, with about 15% lower rejection when the single top is simulated at NLO. Similarly to the fragmentation modeling, an adversarial network is also trained to reduce the sensitivity to factorization scale variations. Since the scale variation is now continuous, the adversary is trained using the mean squared error:

$$L[f, g] = -\sum_{\mu} \Bigg[\Bigg(\sum_{i \in \text{LO single top}} w_i(\mu) \log(f(x_i)) - \sum_{i \in \text{LO } W\text{+jets}} w_i(\mu) \log(1 - f(x_i)) \Bigg) + \lambda \sum_{i \in \text{LO single top}} w_i(\mu) \left(g(f(x_i), y_i) - \mu\right)^2 \Bigg], \tag{3.2}$$

where f is the classifier, g is the adversary, w are weights, and μ is the relative factorization scale. For each event, we can vary the factorization scale through per-event weights w_i and we use values $\mu \in \{0.5, 0.6 \ldots, 1.9, 2\}$ for each event. The adversarially trained classifier is therefore required to reduce the difference in its performance between samples coming from this entire range of scale variations. All hyperparameters are the same as for the fragmentation modeling example shown in the previous section. The performance of the adversarially trained classifier is shown in the bottom plot of Fig. 6. The overall performance is reduced by about a factor of 2 and the sensitivity to factorization scale variations is also significantly reduced by a factor of two or more. While the narrower uncertainty bands may give the impression that the uncertainty has been reduced, in truth the difference between the LO and NLO curves is about the same or bigger than in the nominal case. This means that the 'true' uncertainty would be significantly underestimated using the adversarially trained approach.

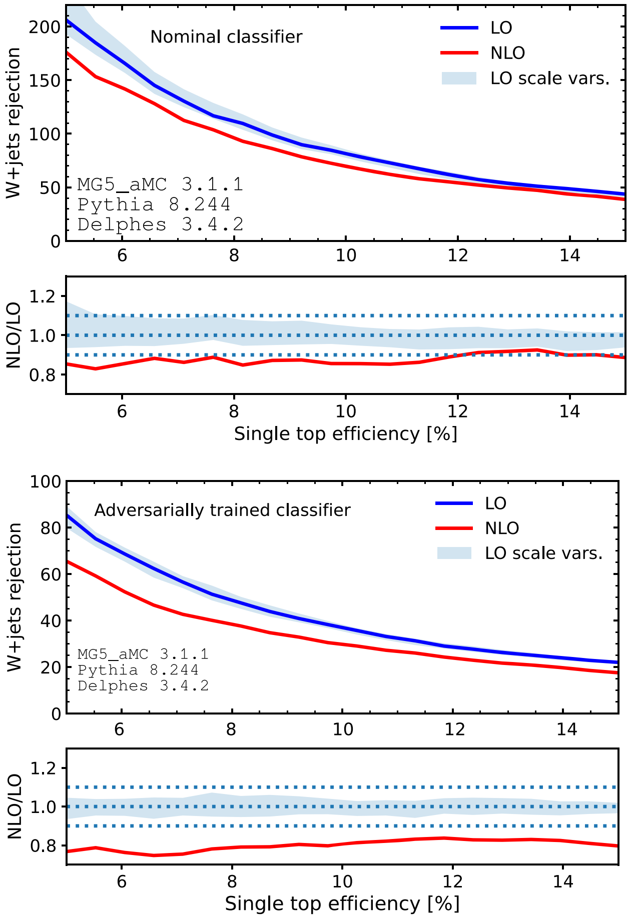

Fig. 6 Top: W+jets rejection (inverse W+jets efficiency) as a function of t-channel single top efficiency for a nominal classifier. The blue band represents the uncertainty estimated by varying the factorization scale by $\frac{1}{2}$ and 2 at LO. Bottom: the same as the top, but for the adversarially trained classifier. Adversarial training only reduces the difference in performance to factorization scale variations, not the difference to NLO, indicating that adversarial training provides a reduced *estimate* of the true uncertainty, which does not translate to a reduction in the true uncertainty

A curious reader may again wonder why the NLO curve does not lie within the uncertainty band coming from scale variations even for the nominal classifier. As in the previous study, the uncertainty bands do not reflect the maximum possible uncertainty but should rather be interpreted as a probabilistic estimate. A study of whether these bands estimates correctly the frequency of higher order computations lying within these bounds is left for a future study. In addition, for this particular example the focus was only on factorization scale variations. This study reveals how decorrelation reduces only the estimate of the uncertainty from scale variations and this does not necessarily translate to actually reducing the difference to NLO.

4 Conclusions and outlook

Decorrelation is a powerful tool for ensuring that machine learning classifiers can be used in practice to enhance analysis sensitivity. However, this tool must be used with caution. We have shown that decorrelation methods may result in significantly underestimated theory uncertainties when using standard approaches to theory uncertainty estimation. In the cases we explored, the estimated uncertainty uses two samples while the ‘true’ uncertainty relies on a third sample that is not part of the training. One could potentially incorporate the third sample into the decorrelation procedure, but there will always be another variation that is not part of the training as long as the full theory uncertainty decomposition is not known. Until we know the complete set of theory nuisance parameters, it seems prudent to not decorrelate away these uncertainties.

While this paper explicitly studied the case for decorrelation, this cautionary tale remains relevant for other uncertainty or inference aware machine learning approaches [9,10,20,73–82] if they are being considered for such theory uncertainties.

Software and data

The software and samples for this paper can be found at https://github.com/hep-lbdl/TheoryUncertDecorrelation.

Acknowledgements We are grateful to Yi-Lun Chung for producing the fragmentation variation samples used in Sect. 3.1. We thank Kingman Cheung, Shih-Chieh Hsu, Tilman Plehn, David Rousseau, David Shih, Michael Spannowsky, and Daniel Whiteson for useful discussions and comments on the manuscript. BN was supported by the Department of Energy, Office of Science under contract number DE-AC02-05CH11231. AG was supported by the U.S. Department of Energy (DOE), Office of Science under Grant no. DE-SC0009920.

Data Availability Statement This manuscript has associated data in a data repository. [Authors’ comment: These data can be found in the Github repository linked above.]

Funded by SCOAP^3.

Appendix A: Training with $\lambda = 0$

Figures 7 and 8 show the impact of using the adversarial setup, but with $\lambda = 0$, i.e. the adversary is turned off. The only difference with respect to the nominal configuration is that Pythia and Herwig (factorization scale variations) are used instead of just Pythia ($\mu = 1$) for the nominal for the two-point (continuous) uncertainty example.

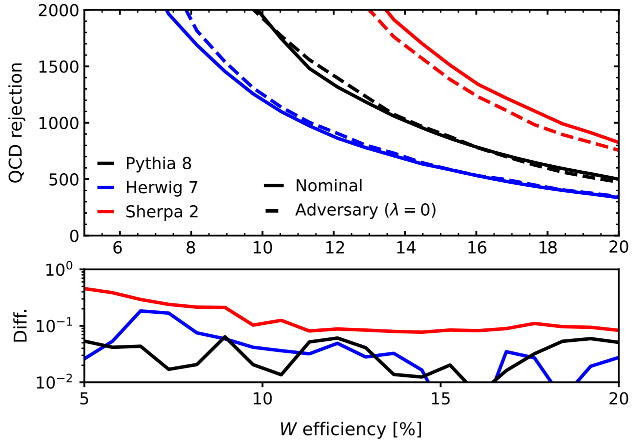

Fig. 7 QCD rejection (inverse QCD efficiency) as a function of W efficiency (study described in Sect. 3.1) for the nominal classifier and for an adversarially trained classifier with $\lambda = 0$. The lower panel is the absolute relative difference for each sample between the nominal and adversarially trained classifiers

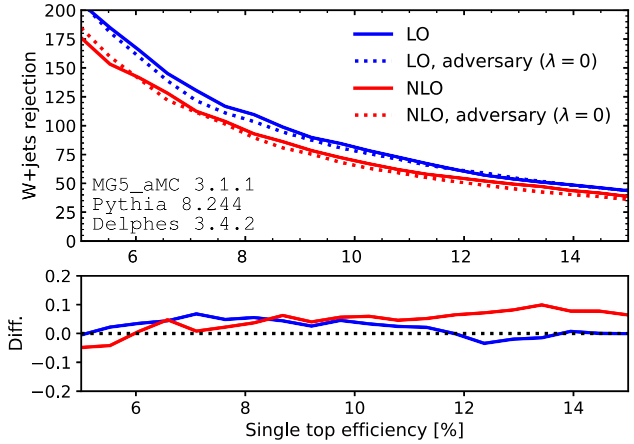

Fig. 8 W+jets rejection (inverse W+jets efficiency) as a function of t-channel single top efficiency (study described in Sect. 3.2) for a nominal classifier (solid lines) and for an adversarially trained classifier (dotted lines) with $\lambda = 0$. The performance on LO samples is shown in blue and on NLO samples in red. The lower panel is the relative difference for LO and NLO samples between the nominal and adversarially trained classifiers

References

1. 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). https://doi.org/10.1016/j.physrep.2019.11.001 arXiv:1709.04464
2. D. Guest, K. Cranmer, D. Whiteson, Deep learning and its application to LHC physics. Annu. Rev. Nucl. Part. Sci. **68**, 161–181 (2018). https://doi.org/10.1146/annurev-nucl-101917-021019 arXiv:1806.11484
3. K. Albertsson et al., Machine learning in high energy physics community white paper. arXiv:1807.02876

4. A. Radovic, M. Williams, D. Rousseau, M. Kagan, D. Bonacorsi, A. Himmel et al., Machine learning at the energy and intensity frontiers of particle physics. Nature **560**, 41–48 (2018). https://doi.org/10.1038/s41586-018-0361-2
5. G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby et al., Machine learning and the physical sciences. Rev. Mod. Phys. **91**, 045002 (2019). https://doi.org/10.1103/RevModPhys.91.045002 arXiv:1903.10563
6. D. Bourilkov, Machine and deep learning applications in particle physics. Int. J. Mod. Phys. A **34**, 1930019 (2020). https://doi.org/10.1142/S0217751X19300199 arXiv:1912.08245
7. M.D. Schwartz, Modern machine learning and particle physics. arXiv:2103.12226
8. M. Feickert, B. Nachman, A living review of machine learning for particle physics. arXiv:2102.02770
9. B. Nachman, A guide for deploying Deep Learning in LHC searches: how to achieve optimality and account for uncertainty. SciPost Phys. **8**, 090 (2020). https://doi.org/10.21468/SciPostPhys.8.6.090 arXiv:1909.03081
10. A. Ghosh, B. Nachman, D. Whiteson, Uncertainty aware learning for high energy physics. Phys. Rev. D **104**, 056026 (2021)
11. K. Cranmer, J. Pavez, G. Louppe, Approximating likelihood ratios with calibrated discriminative classifiers. arXiv:1506.02169
12. P. Baldi, K. Cranmer, T. Faucett, P. Sadowski, D. Whiteson, Parameterized neural networks for high-energy physics. Eur. Phys. J. C **76**, 235 (2016). https://doi.org/10.1140/epjc/s10052-016-4099-4 arXiv:1601.07913
13. J. Dolen, P. Harris, S. Marzani, S. Rappoccio, N. Tran, Thinking outside the ROCs: designing decorrelated taggers (DDT) for jet substructure. JHEP **05**, 156 (2016). https://doi.org/10.1007/JHEP05(2016)156 arXiv:1603.00027
14. I. Moult, B. Nachman, D. Neill, Convolved substructure: analytically decorrelating jet substructure observables. JHEP **05**, 002 (2018). https://doi.org/10.1007/JHEP05(2018)002 arXiv:1710.06859
15. J. Stevens, M. Williams, uBoost: a boosting method for producing uniform selection efficiencies from multivariate classifiers. JINST **8**, P12013 (2013). https://doi.org/10.1088/1748-0221/8/12/P12013 arXiv:1305.7248
16. C. Shimmin, P. Sadowski, P. Baldi, E. Weik, D. Whiteson, E. Goul et al., Decorrelated jet substructure tagging using adversarial neural networks. Phys. Rev. D **96**, 074034 (2017). https://doi.org/10.1103/PhysRevD.96.074034 arXiv:1703.03507
17. L. Bradshaw, R.K. Mishra, A. Mitridate, B. Ostdiek, Mass agnostic jet taggers. SciPost Phys. **8**, 011 (2020). https://doi.org/10.21468/SciPostPhys.8.1.011 arXiv:1908.08959
18. ATLAS Collaboration, Performance of mass-decorrelated jet substructure observables for hadronic two-body decay tagging in ATLAS. Technical Report. ATL-PHYS-PUB-2018-014, CERN, Geneva (2018)
19. G. Kasieczka, D. Shih, Robust jet classifiers through distance correlation. Phys. Rev. Lett. **125**, 122001 (2020). https://doi.org/10.1103/PhysRevLett.125.122001 arXiv:2001.05310
20. L.-G. Xia, QBDT, a new boosting decision tree method with systematical uncertainties into training for High Energy Physics. Nucl. Instrum. Methods A **930**, 15–26 (2019). https://doi.org/10.1016/j.nima.2019.03.088 arXiv:1810.08387
21. A. Rogozhnikov, A. Bukva, V. Gligorov, A. Ustyuzhanin, M. Williams, New approaches for boosting to uniformity. JINST **10**, T03002 (2015). https://doi.org/10.1088/1748-0221/10/03/T03002 arXiv:1410.4140
22. CMS Collaboration, A deep neural network to search for new long-lived particles decaying to jets. Mach. Learn. Sci. Technol. (2020). https://doi.org/10.1088/2632-2153/ab9023. arXiv:1912.12238
23. G. Kasieczka, B. Nachman, M.D. Schwartz, D. Shih, Automating the ABCD method with machine learning. Phys. Rev. D **103**, 035021 (2021). https://doi.org/10.1103/PhysRevD.103.035021 arXiv:2007.14400
24. O. Kitouni, B. Nachman, C. Weisser, M. Williams, Enhancing searches for resonances with machine learning and moment decomposition. JHEP **21**, 070 (2020). https://doi.org/10.1007/JHEP04(2021)070 arXiv:2010.09745
25. G. Louppe, M. Kagan, K. Cranmer, Learning to pivot with adversarial networks. Adv. Neural Inf. Process. Syst. **30**, 981 (2017). arXiv:1611.01046
26. C. Englert, P. Galler, P. Harris, M. Spannowsky, Machine learning uncertainties with adversarial neural networks. Eur. Phys. J. C **79**, 4 (2019). https://doi.org/10.1140/epjc/s10052-018-6511-8 arXiv:1807.08763
27. S. Wunsch, S. Jörger, R. Wolf, G. Quast, Reducing the dependence of the neural network function to systematic uncertainties in the input space. Comput. Softw. Big Sci. **4**, 5 (2020). https://doi.org/10.1007/s41781-020-00037-9 arXiv:1907.11674
28. J.M. Clavijo, P. Glaysher, J.M. Katzy, Adversarial domain adaptation to reduce sample bias of a high energy physics classifier. Mach. Learn. Sci. Tech. **3**(1), 015014 (2022)
29. B. Andersson, G. Gustafson, G. Ingelman, T. Sjostrand, Parton fragmentation and string dynamics. Phys. Rep. **97**, 31–145 (1983). https://doi.org/10.1016/0370-1573(83)90080-7
30. T. Sjostrand, Jet fragmentation of nearby partons. Nucl. Phys. B **248**, 469–502 (1984). https://doi.org/10.1016/0550-3213(84)90607-2
31. T. Sjostrand, S. Mrenna, P.Z. Skands, PYTHIA 6.4 physics and manual. JHEP **05**, 026 (2006). https://doi.org/10.1088/1126-6708/2006/05/026 arXiv:hep-ph/0603175
32. T. Sjostrand, S. Mrenna, P.Z. Skands, A brief introduction to PYTHIA 8.1. Comput. Phys. Commun. **178**, 852–867 (2008). https://doi.org/10.1016/j.cpc.2008.01.036 arXiv:0710.3820
33. B.R. Webber, A QCD model for jet fragmentation including soft gluon interference. Nucl. Phys. B **238**, 492–528 (1984). https://doi.org/10.1016/0550-3213(84)90333-X
34. J.-C. Winter, F. Krauss, G. Soff, A modified cluster hadronization model. Eur. Phys. J. C **36**, 381–395 (2004). https://doi.org/10.1140/epjc/s2004-01960-8 arXiv:hep-ph/0311085
35. J. Bellm et al., Herwig 7.0/Herwig++ 3.0 release note. Eur. Phys. J. C **76**, 196 (2016). https://doi.org/10.1140/epjc/s10052-016-4018-8 arXiv:1512.01178
36. M. Bahr et al., Herwig++ physics and manual. Eur. Phys. J. C **58**, 639–707 (2008). https://doi.org/10.1140/epjc/s10052-008-0798-9 arXiv:0803.0883
37. S. Chang, T. Cohen, B. Ostdiek, What is the machine learning? Phys. Rev. D **97**, 056009 (2018). https://doi.org/10.1103/PhysRevD.97.056009 arXiv:1709.10106
38. L. de Oliveira, M. Kagan, L. Mackey, B. Nachman, A. Schwartzman, Jet-images—deep learning edition. JHEP **07**, 069 (2016). https://doi.org/10.1007/JHEP07(2016)069. arXiv:1511.05190
39. G.J. Székely, M.L. Rizzo, N.K. Bakirov, Measuring and testing dependence by correlation of distances. Ann. Stat. **35**, 2769–2794 (2007). https://doi.org/10.1214/009053607000000505
40. G.J. Székely, M.L. Rizzo, Brownian distance covariance. Ann. Appl. Stat. **3**, 1236–1265 (2009). https://doi.org/10.1214/09-AOAS312
41. G.J. Székely, M.L. Rizzo, The distance correlation t-test of independence in high dimension. J. Multivar. Anal. **117**, 193–213 (2013). https://doi.org/10.1016/j.jmva.2013.02.012
42. G.J. Székely, M.L. Rizzo, Partial distance correlation with methods for dissimilarities. Ann. Stat. **42**, 2382–2412 (2014). https://doi.org/10.1214/14-AOS1255
43. F. Chollet, Keras (2017). https://github.com/fchollet/keras
44. M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean et al., Tensorflow: a system for large-scale machine learning, in *OSDI*, vol. 16 (2016), pp. 265–283

45. D. Kingma, J. Ba, Adam: a method for stochastic optimization. arXiv:1412.6980
46. ATLAS Collaboration, M. Aaboud et al., Performance of top-quark and W-boson tagging with ATLAS in Run 2 of the LHC. Eur. Phys. J. C **79**, 375 (2019). https://doi.org/10.1140/epjc/s10052-019-6847-8. arXiv:1808.07858
47. CMS Collaboration, A.M. Sirunyan et al., Identification of heavy, energetic, hadronically decaying particles using machine-learning techniques. JINST **15**, P06005 (2020). https://doi.org/10.1088/1748-0221/15/06/P06005. arXiv:2004.08262
48. A. Butter et al., The machine learning landscape of top taggers. SciPost Phys. **7**, 014 (2019). https://doi.org/10.21468/SciPostPhys.7.1.014 arXiv:1902.09914
49. J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer et al., 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). https://doi.org/10.1007/JHEP07(2014)079 arXiv:1405.0301
50. R.D. Ball et al., Parton distributions with LHC data. Nucl. Phys. B **867**, 244–289 (2013). https://doi.org/10.1016/j.nuclphysb.2012.10.003 arXiv:1207.1303
51. T. Gleisberg, S. Hoeche, F. Krauss, M. Schonherr, S. Schumann, F. Siegert et al., Event generation with SHERPA 1.1. JHEP **02**, 007 (2009). https://doi.org/10.1088/1126-6708/2009/02/007 arXiv:0811.4622
52. Sherpa Collaboration, E. Bothmann et al., Event generation with Sherpa 2.2. SciPost Phys. **7**, 034 (2019). https://doi.org/10.21468/SciPostPhys.7.3.034. arXiv:1905.09127
53. N. Dawe, E. Rodrigues, H. Schreiner, B. Ostdiek, D. Kalinkin, M.R. et al., scikit-hep/pyjet: version 1.8.2. Jan. (2021). https://doi.org/10.5281/zenodo.4446849
54. M. Cacciari, G.P. Salam, G. Soyez, FastJet user manual. Eur. Phys. J. C **72**, 1896 (2012). https://doi.org/10.1140/epjc/s10052-012-1896-2 arXiv:1111.6097
55. M. Cacciari, G.P. Salam, G. Soyez, The anti-k_t jet clustering algorithm. JHEP **04**, 063 (2008). https://doi.org/10.1088/1126-6708/2008/04/063 arXiv:0802.1189
56. J. Thaler, K. Van Tilburg, Identifying boosted objects with N-subjettiness. JHEP **03**, 015 (2011). https://doi.org/10.1007/JHEP03(2011)015 arXiv:1011.2268
57. J. Thaler, K. Van Tilburg, Maximizing boosted top identification by minimizing N-subjettiness. JHEP **02**, 093 (2012). https://doi.org/10.1007/JHEP02(2012)093 arXiv:1108.2701
58. A.J. Larkoski, I. Moult, D. Neill, Power counting to better jet observables. JHEP **12**, 009 (2014). https://doi.org/10.1007/JHEP12(2014)009 arXiv:1409.6298
59. A.J. Larkoski, G.P. Salam, J. Thaler, Energy correlation functions for jet substructure. JHEP **06**, 108 (2013). https://doi.org/10.1007/JHEP06(2013)108 arXiv:1305.0007
60. Particle Data Group, Review of particle physics. Prog. Theor. Exp. Phys. **2020**, 08 (2020). https://doi.org/10.1093/ptep/ptaa104
61. J.M. Butterworth, A.R. Davison, M. Rubin, G.P. Salam, Jet substructure as a new Higgs search channel at the LHC. Phys. Rev. Lett. **100**, 242001 (2008). https://doi.org/10.1103/PhysRevLett.100.242001 arXiv:0802.2470
62. S.D. Ellis, C.K. Vermilion, J.R. Walsh, Recombination algorithms and jet substructure: pruning as a tool for heavy particle searches. Phys. Rev. D **81**, 094023 (2010). https://doi.org/10.1103/PhysRevD.81.094023 arXiv:0912.0033
63. D. Krohn, J. Thaler, L.-T. Wang, Jet trimming. JHEP **02**, 084 (2010). https://doi.org/10.1007/JHEP02(2010)084 arXiv:0912.1342
64. M. Dasgupta, A. Fregoso, S. Marzani, G.P. Salam, Towards an understanding of jet substructure. JHEP **09**, 029 (2013). https://doi.org/10.1007/JHEP09(2013)029 arXiv:1307.0007
65. A.J. Larkoski, S. Marzani, G. Soyez, J. Thaler, Soft drop. JHEP **05**, 146 (2014). https://doi.org/10.1007/JHEP05(2014)146 arXiv:1402.2657
66. Y. Ganin, V. Lempitsky, Unsupervised domain adaptation by backpropagation. Proc. Mach. Learn. Res. **37**, 1180–1189 (2015)
67. ATLAS Collaboration, M. Aaboud et al., Measurement of the inclusive cross-sections of single top-quark and top-antiquark t-channel production in pp collisions at $\sqrt{s}$ = 13 TeV with the ATLAS detector. JHEP **04**, 086 (2017). https://doi.org/10.1007/JHEP04(2017)086. arXiv:1609.03920
68. CMS Collaboration, A.M. Sirunyan et al., Measurement of differential cross sections and charge ratios for t-channel single top quark production in proton–proton collisions at $\sqrt{s}$ = 13 TeV. Eur. Phys. J. C **80**, 370 (2020). https://doi.org/10.1140/epjc/s10052-020-7858-1. arXiv:1907.08330
69. DELPHES 3 Collaboration, J. de Favereau, C. Delaere, P. Demin, A. Giammanco, V. Lemaitre, A. Mertens et al., DELPHES 3, a modular framework for fast simulation of a generic collider experiment. JHEP **02**, 057 (2014). https://doi.org/10.1007/JHEP02(2014)057. arXiv:1307.6346
70. A. Mertens, New features in Delphes 3. J. Phys. Conf. Ser. **608**, 012045 (2015). https://doi.org/10.1088/1742-6596/608/1/012045
71. M. Selvaggi, DELPHES 3: a modular framework for fast-simulation of generic collider experiments. J. Phys. Conf. Ser. **523**, 012033 (2014). https://doi.org/10.1088/1742-6596/523/1/012033
72. M. Cacciari, G.P. Salam, Dispelling the N^3 myth for the k_t jet-finder. Phys. Lett. B **641**, 57 (2006). https://doi.org/10.1016/j.physletb.2006.08.037 arXiv:hep-ph/0512210
73. S. Wunsch, S. Jörger, R. Wolf, G. Quast, Optimal statistical inference in the presence of systematic uncertainties using neural network optimization based on binned Poisson likelihoods with nuisance parameters. Comput. Softw. Big Sci. **5**, 4 (2021). https://doi.org/10.1007/s41781-020-00049-5 arXiv:2003.07186
74. A. Elwood, D. Krücker, M. Shchedrolosiev, Direct optimization of the discovery significance in machine learning for new physics searches in particle colliders. J. Phys. Conf. Ser. **1525**, 012110 (2020). https://doi.org/10.1088/1742-6596/1525/1/012110
75. P. De Castro, T. Dorigo, INFERNO: inference-aware neural optimisation. Comput. Phys. Commun. **244**, 170–179 (2019). https://doi.org/10.1016/j.cpc.2019.06.007 arXiv:1806.04743
76. T. Charnock, G. Lavaux, B.D. Wandelt, Automatic physical inference with information maximizing neural networks. Phys. Rev. D **97** (2018). https://doi.org/10.1103/physrevd.97.083004
77. J. Alsing, B. Wandelt, Nuisance hardened data compression for fast likelihood-free inference. Mon. Not. R. Astron. Soc. **488**, 5093–5103 (2019). https://doi.org/10.1093/mnras/stz1900 arXiv:1903.01473
78. L. Heinrich, N. Simpson, pyhf/neos: initial zenodo release (2020). https://doi.org/10.5281/zenodo.3697981
79. J. Brehmer, F. Kling, I. Espejo, K. Cranmer, MadMiner: machine learning-based inference for particle physics. Comput. Softw. Big Sci. **4**, 3 (2020). https://doi.org/10.1007/s41781-020-0035-2 arXiv:1907.10621
80. J. Brehmer, G. Louppe, J. Pavez, K. Cranmer, Mining gold from implicit models to improve likelihood-free inference. Proc. Natl. Acad. Sci. 201915980 (2020). https://doi.org/10.1073/pnas.1915980117. arXiv:1805.12244
81. J. Brehmer, K. Cranmer, G. Louppe, J. Pavez, Constraining effective field theories with machine learning. Phys. Rev. Lett. **121**, 111801 (2018). arXiv:1805.00013
82. J. Brehmer, K. Cranmer, G. Louppe, J. Pavez, A guide to constraining effective field theories with machine learning. Phys. Rev. D **98**, 052004 (2018). https://doi.org/10.1103/PhysRevD.98.052004 arXiv:1805.00020