Title
Uncertainty-aware machine learning for high energy physics

Permalink
https://escholarship.org/uc/item/9gc7h8p0

Journal
Physical Review D, 104(5)

ISSN
2470-0010

Authors
Ghosh, Aishik
Nachman, Benjamin
Whiteson, Daniel

Publication Date
2021-09-01

DOI
10.1103/physrevd.104.056026

Peer reviewed

Uncertainty-aware machine learning for high energy physics

Aishik Ghosh,[1,2] Benjamin Nachman,[2,3] and Daniel Whiteson[1]

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

(Received 3 June 2021; accepted 28 August 2021; published 28 September 2021)

Machine learning techniques are becoming an integral component of data analysis in high energy physics. These tools provide a significant improvement in sensitivity over traditional analyses by exploiting subtle patterns in high-dimensional feature spaces. These subtle patterns may not be well modeled by the simulations used for training machine learning methods, resulting in an enhanced sensitivity to systematic uncertainties. Contrary to the traditional wisdom of constructing an analysis strategy that is invariant to systematic uncertainties, we study the use of a classifier that is fully aware of uncertainties and their corresponding nuisance parameters. We show that this dependence can actually enhance the sensitivity to parameters of interest. Studies are performed using a synthetic Gaussian dataset as well as a more realistic high energy physics dataset based on Higgs boson decays to tau leptons. For both cases, we show that the uncertainty aware approach can achieve a better sensitivity than alternative machine learning strategies.

DOI: 10.1103/PhysRevD.104.056026

I. INTRODUCTION

The usefulness of physical measurements is tied to the magnitude and reliability of their estimated uncertainties. Whether one is measuring the mass of the Higgs boson [1] or the value of the Hubble constant [2], it is the size of the uncertainty that communicates the quality of the information and allows measurements to be contrasted or accurately combined. While statistical uncertainties can be reduced with the collection of additional data, the most troublesome type of uncertainty is systematic uncertainty. These uncertainties can arise from many sources and are often modeled as the dependence of a parameter of interest on other degrees of freedom known as "nuisance parameters."

The challenging task of quantifying the systematic uncertainty on a measured physical parameter has become even more important due to the growing use of complex statistical procedures based on modern machine learning [3–9]. While more powerful techniques can extract more information from higher-dimensional datasets, they may also introduce or enhance the dependence of those measurements on nuisance parameters. This is because, despite their importance, systematic uncertainties are often not part of the learning procedure for a typical machine learning model. Instead, models are typically trained on synthetic datasets generated with assumed values of the nuisance parameters; the impact of uncertainties is typically quantified *post hoc* by varying those nuisance parameters. In cases where systematic uncertainties are dominant, such machine learning models may improve the statistical uncertainty but increase the systematic uncertainty by exacerbating the dependence on nuisance parameters. To minimize the total uncertainty, it is essential to incorporate uncertainties directly into the learning procedure.

To date, several approaches have been considered. Data augmentation trains a model on a concoction of synthetic data with different values of the nuisance parameters. This exposes the classifiers to the range of possible nuisance parameter values, so that at inference time, the result may be less sensitive to their precise value. The disadvantage of augmentation is that the model is unaware of the values of the nuisance parameters, and so learns the average, rather than the optimal response. Another possibility is to train a model to explicitly be insensitive to nuisance parameters or other parameters [10–26]. One implementation of this approach involves training two machine learning models at the same time: one that achieves the primary learning task and a second model that tries to learn information about the nuisance parameter/feature from the output of the classifier. When this second model (adversary) is unable to perform its task, then the primary classifier is insensitive, as desired. The data augmentation and adversarial learning approaches will serve as important baselines in this paper.

Rendering a classifier insensitive to nuisance parameters may increase analysis precision and decrease analysis complexity, but it may also have undesirable consequences. First, systematic uncertainties often involve guesswork; in many cases, the corresponding nuisance parameters do not have a strict statistical origin. Reducing the sensitivity to a particular nuisance parameter may not eliminate the underlying uncertainty; instead, it may eliminate the only existing handle to probe the source of uncertainty. For example, a measurement which used a model constructed to be insensitive to differences between two example parton shower algorithms, such as PYTHIA [27] and Herwig [28], may still have significant systematic uncertainty due to our lack of a complete understanding of fragmentation. Second, some values of the nuisance parameter achieve a better sensitivity to the parameters of interest than others and this could be exploited to improve the performance of a classifier. A classifier that is insensitive to a nuisance parameter will not be able to exploit features that are highly sensitive to that parameter. The first example in this paper demonstrates an extreme case where a classifier insensitive to a nuisance parameter will result in no separation power.

We advocate for the opposite of decorrelation. Classifiers are constructed to be explicitly dependent on nuisance parameters as if they are parameters of interest. As nuisance parameters are profiled, the classifier will change and the best classifier will be used for each value of the nuisance parameter. Parametrized classifiers have been studied in the context of parameters or features of interest [29,30], and full dependence on nuisance parameters for inference has been advocated in Refs. [31–35].

In this paper, we provide specific examples of profiled classifiers and show explicitly that they can enhance analysis sensitivity over strategies that render networks insensitive to nuisance parameters. We focus on only the construction of classifiers as useful statistics for downstream analysis and not on full likelihood (ratio) estimation. In this way, our uncertainty-aware classifier approach is a straightforward extension of existing analyses performed at the Large Hadron Collider (LHC) and therefore may result in immediate improvements in sensitivity. In addition, this prescription allows for easy *post hoc* histogram-based diagnostics. These may include quantification of the impact of additional sources of systematic uncertainties that are not used for training, and checks for whether the measurement overconstrains the nuisance parameter.

While we focus on the profiling aspect of uncertainty awareness, there is a complementary line of research on the use of uncertainty-aware loss functions [36–42]. These approaches construct classifiers that are optimized using the final test statistic, including uncertainties. We leave the combination of profile-aware training and uncertainty-aware loss to future work. There have also been recent proposals to use Bayesian neural networks for estimating uncertainties [43–46]. Additional information about the interplay between uncertainties and machine learning can be found in recent reviews [35,47].

This paper is organized as follows. The uncertainty-aware methods are introduced in Sec. II. Evaluation criteria for assessing the performance of the methods is described in Sec. III. To build intuition, a Gaussian example is presented in Sec. IV. A physics example based on Higgs boson decays to τ leptons using a benchmark dataset is provided in Sec. V. The paper ends with conclusions and outlook in Sec. VI.

II. UNCERTAINTY-AWARE METHODS

This section describes the four methods of training classifiers studied in this paper. All neural networks were trained using KERAS 2.2.4 [48] with a TensorFlow 1.12.0 [49] backend on a single Nvidia GeForce GTX GPU.

A. Notation

Features used for classification are denoted by $X \in \mathbb{R}^n$, where lower case x is a realization of the random variable X. A dataset of many examples will be written $\{x_i\}$. The nominal value of the nuisance parameter $z \in \mathbb{R}^m$ is z_0 and the true value is z_T. In some cases, we will promote the nuisance parameters to random variables, in which case they will be represented by the capital letter Z.

B. Baseline classifier

The baseline method is a classifier trained to distinguish signal and background using data simulated at the nominal value of the nuisance parameter, as is done routinely in LHC analyses. A network trained optimally to minimize a binary cross-entropy (BCE) loss learns to output a score (see, e.g., [50]),

$$s(x) = \frac{p(x|z = z_0, S)}{p(x|z = z_0, S) + p(x|z = z_0, B)}, \tag{1}$$

where $p(\cdot)$ denotes a probability density, S represents the signal class and B represents the background class. The score of the network is used as an observable with high sensitivity to the parameter of interest for the final measurement.

C. Data augmentation

An alternative method is to augment the training data to include signal and background samples with several values of the nuisance parameters. A network trained optimally to minimize a BCE loss learns the score,

$$s(x) = \frac{\langle p(x|Z, S)\rangle_{p_Z}}{\langle p(x|Z, S)\rangle_{p_Z} + \langle p(x|Z, B)\rangle_{p_Z}}, \tag{2}$$

where p_Z is the probability density over the nuisance parameter Z, treated as a random variable with some probability density chosen by the experimenter. Typically, Z is discrete and has a nonzero probability mass at only a few values. The score $s(x)$ is then treated in the same way as in the baseline case [Eq. (1)].

D. Adversarial training

An orthogonal strategy is to train a classifier with the explicit objective of being insensitive to the effects of the nuisance parameter. Our implementation follows the adversarial training prescription of Ref. [12]. However, to improve the training stability and speed, the classifier and adversary are concatenated together through a gradient reversal layer [51] and trained simultaneously. The classifier is trained with the objective to minimize the classification loss and maximize the adversarial loss and the second loss has a relative weight of λ, a tunable hyperparameter.

While training for exact invariance in this adversarial setup can be tricky [52], maximizing overall sensitivity requires a compromise between the level of invariance to nuisance parameters and the classification power. The Gaussian case described in Sec. IV is an extreme example where exact invariance to the nuisance parameter requires zero discriminating power for the classifier.

In the end, the score of the classifier on observed data is used as an observable in the final measurement, in the same way as for the baseline classifier.

E. Uncertainty-aware classifier

The concept explored in this paper is to parametrize the network in the nuisance parameters, see Fig. 1. Specifically, the network is trained with the true value of the nuisance parameter z as an input to the network in additional to the observables x. A network trained optimally to minimize a BCE loss learns the score,

$$s(x,z) = \frac{p(x|Z=z,S)}{p(x|Z=z,S)+p(x|Z=z,B)}. \tag{3}$$

The score of this classifier is not used as a single observable for the final fit as in the previous methods.

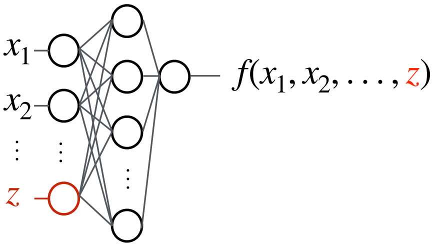


FIG. 1. The architecture of an uncertainty-aware network, in which the nuisance parameter z is treated as a feature alongside the observed data x, learning a decision function which varies with the nuisance parameter.

At evaluation time, while the x values remain fixed as inputs to the network, the unknown z is left as a parameter, allowing for later profiling over the nuisance parameters in the final measurement.

Importantly, note that Eq. (3) depends on z. This means that the calculation of analysis observable(s) depends on z and change as the nuisance parameter is varied, during the evaluation of uncertainties and/or during nuisance parameter profiling. This is in contrast to the standard search paradigm in which the calculation of the analysis observables are fixed and the sensitivity to z is evaluated *post hoc*. Allowing the calculation of the analysis observables to depend explicitly on the value of z is not the traditional approach, but it does not require that the experimenter have any special knowledge of z. Formation of a confidence interval in the space of model parameters (either parameters of interest or nuisance parameters) naturally requires calculating the likelihood ratio of the model as those parameters vary, relative to the best-fit parameters. It is natural for the calculation of the analysis observable, a proxy for the likelihood ratio, to vary with those parameters. One can later profile over the nuisance parameters to capture the impact of our lack of knowledge of its true value. The traditional approach of fixing the analysis observable calculation can be thought of as an *ad hoc* approximation of the full method.

III. EVALUATION METHODOLOGY

To evaluate the power of each approach above, we apply them to a common use case, fitting a signal hypothesis in the presence of background, where both signal and background depend on nuisance parameters. Relevant to many measurements of Standard Model (SM) processes as well as searches for physics beyond the SM, the parameter of interest is the signal strength μ, the cross section of the signal relative to the reference value. In the Gaussian example below, we use low-dimensional datasets for simpler visualization, but the results generalize. Similarly, for ease of calculations we perform a binned likelihood fit, although the unbinned nature of neural networks should allow application to unbinned cases; we leave that investigation to future work.

For each of the strategies described, template histograms of the classifier score are constructed from simulated signal and background events for several values of the nuisance parameter z. These templates are the basis of the binned likelihood calculation $\mathcal{L}(\mu, z|\{x_i\})$ over the parameters μ, z, where $\{x_i\}$ is the full observed dataset. The likelihood is a product of a Poisson term for each histogram bin and a Gaussian constraint on the nuisance parameter. The Gaussian constraint can readily be replaced with any other prior or a Poisson term from an auxiliary measurement if z is directly constrained with control region data (demonstrated in Appendix B). If no additional prior or constraint on the nuisance parameter is used then only information from the primary measurement constrains z. The negative log-likelihood (NLL) is (up to an irrelevant constant),

$$
\begin{aligned}
&-\log \mathcal{L}(\mu, z|\{x_i\}) \\
&\quad = -\sum_{j=1}^{n_{\mathrm{bins}}} [N_j \cdot \log(\mu s_j + b_j) - \mu s_j - b_j - \log(\Gamma(N_i))] \\
&\qquad + \left(\frac{z - z_0}{\sqrt{2}\sigma_z}\right)^2, \qquad (4)
\end{aligned}
$$

where s_j, b_j are the expected number of signal and background events in bin j, respectively, and N_j is the number of events observed in data for that bin. The Γ function is the generalized factorial function that can handle decimal values in the simulated test dataset. Although usually irrelevant, the $\log(\Gamma(N_i))$ term is not a constant while using an uncertainty-aware network and cannot be ignored. For this approach, the decision function changes with z and therefore the bin counts in simulation and observed data also change with z.

In practice, samples at various values of z can often be produced cheaply from a single simulated Monte Carlo sample by shifting the value of z and recomputing all the relevant physics variables, and this approach will be used for the studies in Sec. V. Care must be taken to apply any kinematic selection on these variables only after the shift. In these studies, the templates and the "observed dataset" are built using the same test dataset because the dataset used in Sec. V is not large enough to split into three representative datasets.

The fitted value of μ is obtained by minimizing Eq. (4). Uncertainties are accounted for by studying the dependence of the likelihood near the fitted value $\hat{\mu}$ while optimizing over z. The power of each approach is determined by their relative uncertainties in μ. As a diagnostic, the parameter of interest may be profiled over instead to check if the measurement overconstrains the nuisance parameter.

IV. GAUSSIAN EXAMPLE

To illustrate the different approaches in a simple setting with complete analytic control, we begin with a Gaussian example with a two-dimensional feature space and a single nuisance parameter. Signal events are drawn from Gaussian distributions in the two features, with means at $\cos(z)$ and $\sin(z)$, respectively; the width of each is set to 0.7. Background events are generated in same fashion, but with means for the two features at $-\cos(z)$ and $-\sin(z)$, respectively. An example of the signal and background distributions for $z = \frac{\pi}{4}$ is shown in Fig. 2.

A set of 4.2×10^7 events are generated at 21 values of z equally spaced between 0 and $\pi/2$ for the signal and background. The dataset is split into training and test sets with a ratio of 3:1. All signal events in the test set have a weight of 10^{-3} and all background events have a weight of 10^{-1} to mimic a rare signal typical of LHC analyses. Ten bins are used to construct the template and observed histograms. The parameter of interest is the signal strength μ with a true value of 1.

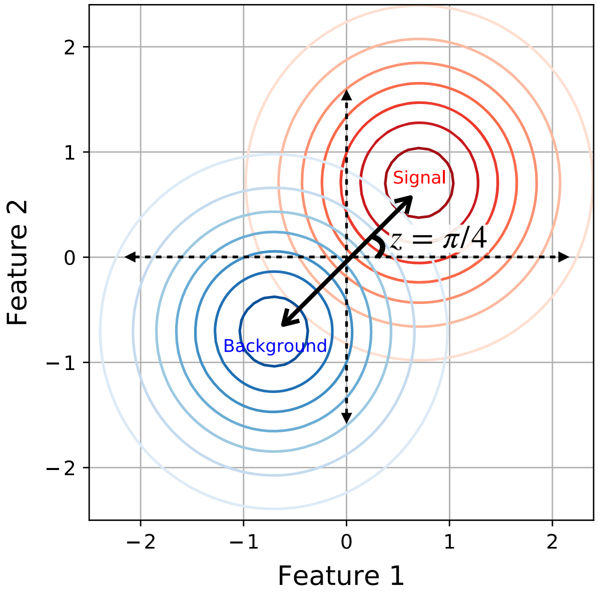


FIG. 2. Contour of probability densities for signal and background hypotheses in the two-dimensional feature space for the simple Gaussian demonstration case, with the nuisance parameter fixed to $z = \frac{\pi}{4}$.

A. Models

In a simple case where the signal and background probabilities are well known, it is possible to derive the classifier analytically for the baseline and uncertainty-aware approaches. The results below use the analytical expressions, but as a cross check, neural networks were also trained for the same objective and produced nearly identical results.

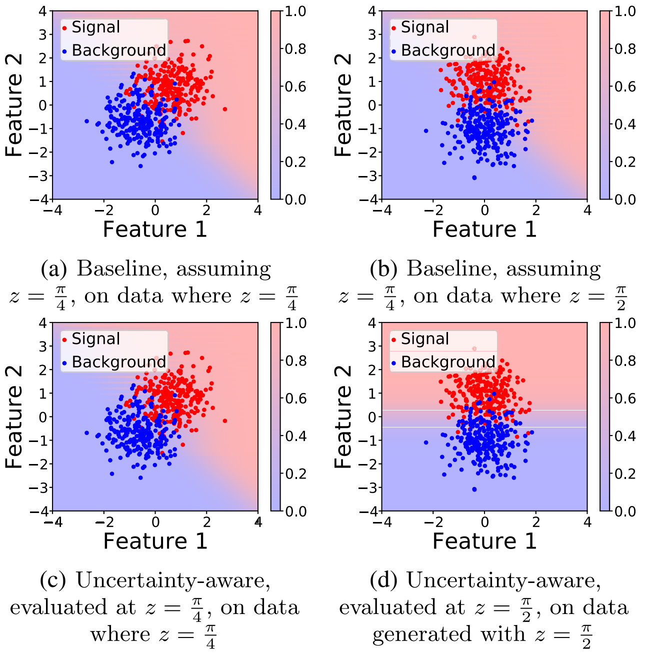
1. Baseline and uncertainty-aware classifiers

The baseline classifier computes the score

$$
s(x) = \frac{p(x|z = \frac{\pi}{4}, S)}{p(x|z = \frac{\pi}{4}, S) + p(x|z = \frac{\pi}{4}, B)}, \qquad (5)
$$

using the probability density functions for the Gaussian distributions used to generate the two features for signal and background at an assumed fixed value of $z = \frac{\pi}{4}$.

The uncertainty-aware classifier, on the other hand, does not make assumptions about the value of the nuisance parameter and instead calculates a score as a function of the nuisance parameter



(a) Baseline, assuming $z = \frac{\pi}{4}$, on data where $z = \frac{\pi}{4}$

(b) Baseline, assuming $z = \frac{\pi}{4}$, on data where $z = \frac{\pi}{2}$

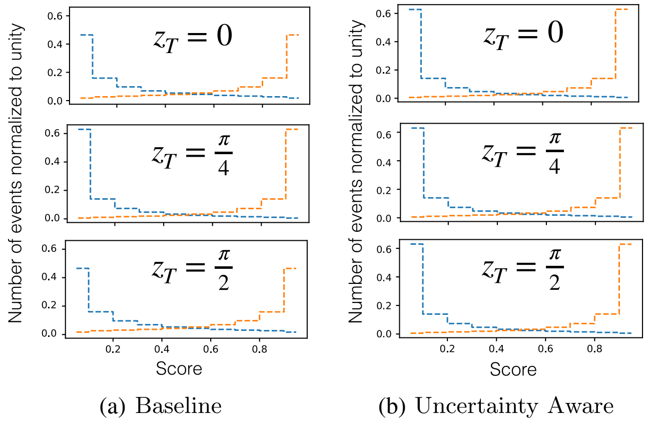
(c) Uncertainty-aware, evaluated at $z = \frac{\pi}{4}$, on data where $z = \frac{\pi}{4}$

(d) Uncertainty-aware, evaluated at $z = \frac{\pi}{2}$, on data generated with $z = \frac{\pi}{2}$

FIG. 3. Classifier score for the baseline and systematic-aware classifiers, see text for definitions. Shown are examples where the baseline classifier's assumption that the nuisance parameter is $z = \frac{\pi}{4}$ matches or disagrees with the generated data (points). Also shown are score functions for the uncertainty-aware classifier on the same datasets, evaluated at the correct value of z for each dataset.

$$s(x,z) = \frac{p(x|Z=z,S)}{p(x|Z=z,S) + p(x|Z=z,B)}. \qquad (6)$$

The score s, for the each of the two classifiers are shown in Fig. 3 as a function of the input features, for datasets generated with $z = \frac{\pi}{4}$ or $z = \frac{\pi}{2}$. Figures 3(a) and 3(b) show that the baseline classifier, which assumes $z = \frac{\pi}{4}$, provides a decision function that is appropriate for data generated at $z = \frac{\pi}{4}$ but not for data generated at $z = \frac{\pi}{4}$. Figures 3(c) and 3(d) show that the uncertainty-aware classifier can provide an appropriate decision function for data generated at either value of z. The uncertainty-aware classifier is parametrized as a function of z, and given any value of the nuisance parameter, it can provide the appropriate classifier. Examples of histogram templates of the classifier outputs are shown in Fig. 4. The separation power of the baseline classifier is clearly reduced for cases where the data are generated with values of the nuisance parameter that do not match its assumed value of $z = \frac{\pi}{4}$. Using the area under the receiver operating characteristic (ROC) curve as a metric to quantify separation power of a model, the separation power for the baseline classifier falls from 0.978 for data generated at $z = \frac{\pi}{4}$ to 0.924 for data generated at $z = \frac{\pi}{2}$, while it remains 0.978 on both datasets for the uncertainty-aware classifier. A comparison of ROC curves is provided in Appendix A (Fig. 13).



(a) Baseline

(b) Uncertainty Aware

FIG. 4. Template histograms of the classifier score for the baseline (left) and uncertainty-aware approaches (right) evaluated for data generated at various true values of z. The signal distribution is shown in orange and the background distribution in blue. The baseline classifier assumes $z = \frac{\pi}{4}$, and loses separation power for data generated with $z = \{0, \frac{\pi}{2}\}$, manifested by the lower heights of the signal and background histograms near 1 and 0, respectively. The uncertainty-aware classifier score is evaluated for the correct value of z, providing the optimal score in each case.

2. Data augmentation

A linear discriminant analysis classifier from Scikit-learn [53] is trained on a training dataset that includes samples with all 21 values[1] of z. As a cross-check, a neural network was trained on the same data and produced a nearly identical score function.

3. Adversarial training

The adversarial architecture was trained using samples from all 21 values of z. The classifier and the adversarial network each consist of 10 hidden layers with 64 nodes and a rectified linear unit (RLU) activation and a single node output layer with sigmoid and linear activations respectively. An L2 kernel regularizer [54] was applied to all but the first and final layer of each network. The two networks were attached with a gradient reversal layer, which scales the gradient by −0.2 and trained with the RMSProp [55] optimizer and a batch size of 4096. BCE is used as the classification loss while mean squared error is used for the loss of the adversary. An adversarial loss weight of $\lambda = 1$ was used. For this dataset, a classifier exactly invariant to z would have zero separation power between signal and background. Therefore, a compromise between invariance

[1] The data augmentation classifier was also trained on a dataset with a continuous distribution of z sampled from the Gaussian prior of z and found to provide near identical results.

and classification power was made in model selection, finding the largest value of λ that did not deteriorate performance. Minimal hyperparameter tuning was performed beyond tuning λ.

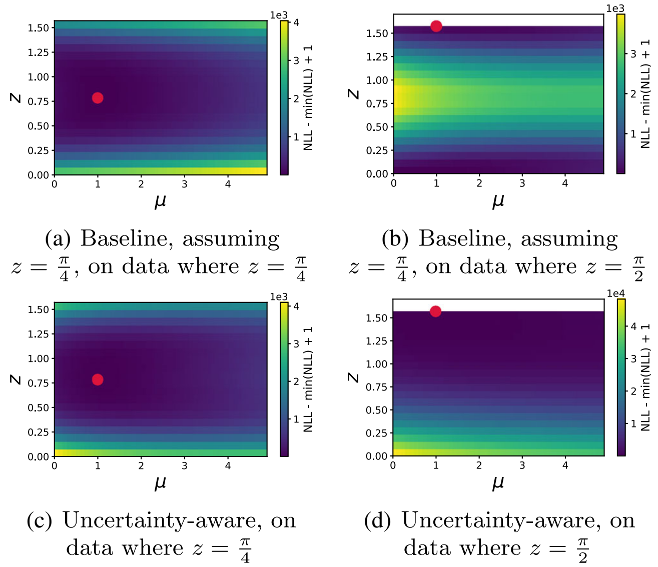
B. Results

The negative log-likelihood [Eq. (4)] is calculated as a function of the parameter of interest μ and the nuisance parameter z. Examples are shown in Fig. 5 using templates from the baseline and uncertainty-aware classifiers. Due to its assumption that $z = \frac{\pi}{4}$ in the calculation of the classifier score, the likelihood from the baseline classifier can strongly exclude $z = \frac{\pi}{4}$ when evaluated on a dataset generated with $z = \frac{\pi}{2}$, but finds $z = 0, \frac{\pi}{2}$ equally likely. The uncertainty-aware classifier, on the other hand, is also able to exclude the low z region.

Since the measurement of the nuisance parameter is not the final objective, it is in fact the profile likelihood, $\mathcal{L}_p(\mu) = \max_z \mathcal{L}(\mu, z)$, that is the most relevant metric for determining the relative power of the various approaches. The dependence of the likelihood on the nuisance parameter is thus profiled away.

The profile likelihood for each method is shown in Fig. 6 for data generated with $z = \frac{\pi}{4}$ and $z = \frac{\pi}{2}$. In the case of $z = \frac{\pi}{4}$, which matches the assumption of the baseline classifier, the uncertainty-aware and baseline classifiers both achieve ideal performance. The adversarial and data-augmentation approaches are somewhat weaker due to the inherent compromises of their methods.

When evaluated on data generated with $z = \frac{\pi}{2}$, in conflict with the assumption of the baseline classifier, the performance of all approaches other than the uncertainty-aware classifier deteriorate significantly. The performance of the data-augmented classifier depends on the range of z values available in training, with performance being strongest near the center of the range; one could shift the range to improve performance at the extreme values. No setting of the adversarially trained classifier was found to perform well for datasets with both values of z.



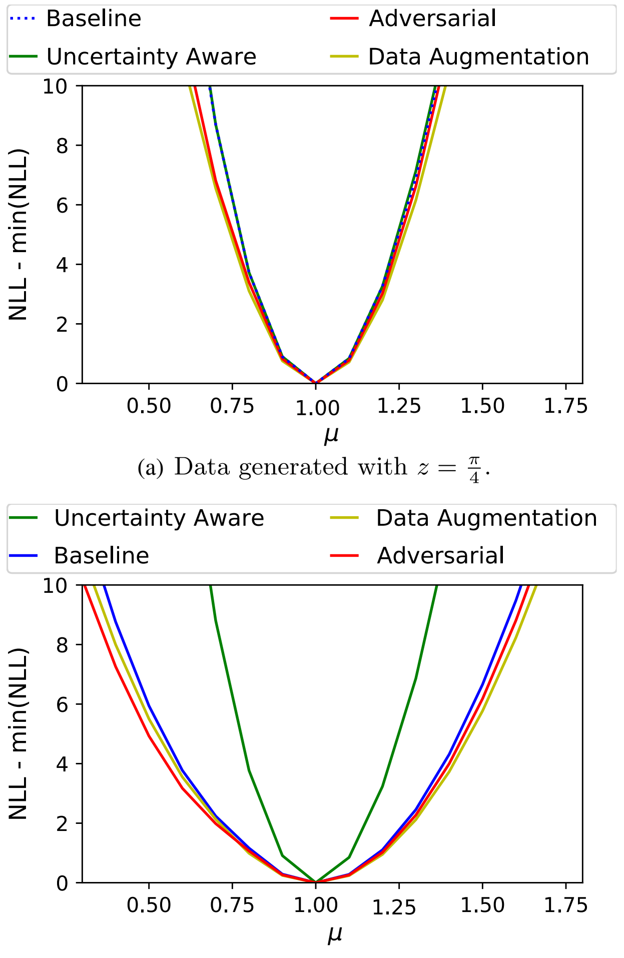
(a) Baseline, assuming $z = \frac{\pi}{4}$, on data where $z = \frac{\pi}{4}$

(b) Baseline, assuming $z = \frac{\pi}{4}$, on data where $z = \frac{\pi}{2}$

(c) Uncertainty-aware, on data where $z = \frac{\pi}{4}$

(d) Uncertainty-aware, on data where $z = \frac{\pi}{2}$

FIG. 5. The negative log-likelihood [Eq. (4)] as a function of the parameter of interest μ and the nuisance parameter z for two example datasets, using templates from the baseline (top) and uncertainty-aware classifier (bottom). In the left column, the data are generated with $z = \frac{\pi}{4}$, which matches the assumption made by the baseline classifier. In the right column, the data are generated with $= \frac{\pi}{2}$. The red dot indicates the maximum likelihood estimate which coincides with the true value of μ, z in each case. Note the different z axis scales for the two classifiers in the bottom row.



(a) Data generated with $z = \frac{\pi}{4}$.

(b) Data generated with $z = \frac{\pi}{2}$.

FIG. 6. The profile likelihood $\max_z \mathcal{L}(\mu, z)$ as a function of the parameter of interest, μ for likelihoods calculated with templates built from the various classifiers. Narrower curves indicate more precise measurements having accounted for systematic and statistical uncertainties. The baseline classifier assumes $z = \frac{\pi}{4}$, and matches the performance of the uncertainty-aware classifier in data generated with $z = \frac{\pi}{4}$ (top). In data generated with $z = \frac{\pi}{2}$, the power of all classifiers other than the uncertainty-aware classifier become significantly weaker.

V. REALISTIC EXAMPLE

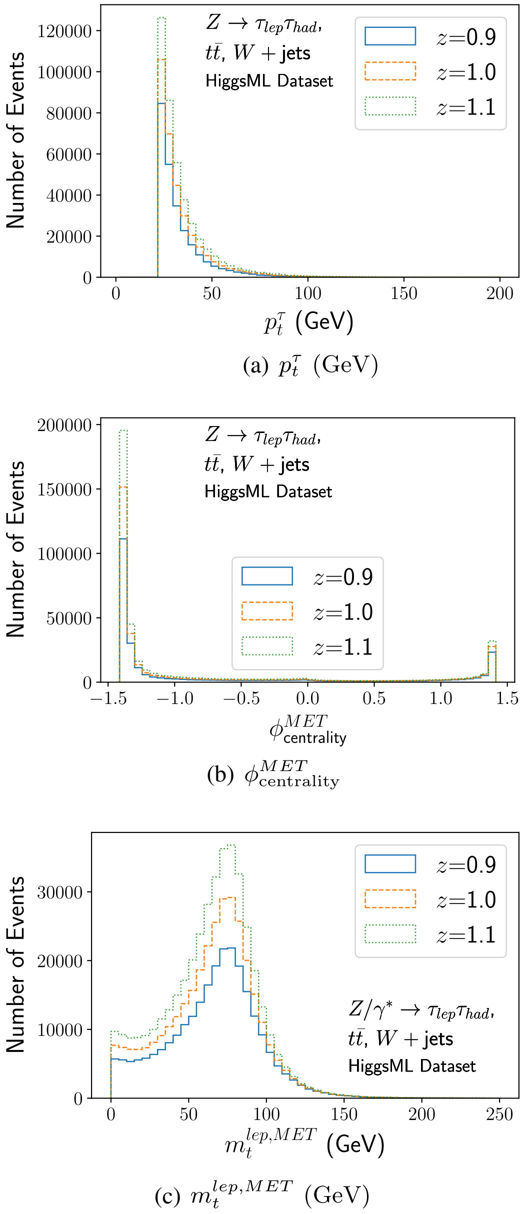
A more realistic application of the uncertainty-aware classifier in the presence of nuisance parameters can be performed using the datasets [56] produced for the HiggsML Kaggle challenge [57] by the ATLAS Collaboration. This dataset was originally simulated by the ATLAS collaboration to measure the decay of the Higgs boson to a pair of τ leptons [58]. This dataset was chosen for our study because it has been used as a benchmark for uncertainty aware learning in the past [52,59].

The signal process is the production of Higgs bosons through gluon-gluon fusion, vector boson fusion, and associated production with a vector boson, which decays to pairs of τ leptons. The gluon-gluon fusion and vector boson fusion production processes were simulated with POWHEG [60–63] interfaced to PYTHIA8 [27] while the vector boson production is simulated with PYTHIA8. Further details on corrections applied can be found in Sec 3. of Ref. [58]. The detector response is simulated with GEANT4 [64] and object reconstruction performed with the official ATLAS software [65]. The three largest backgrounds from $Z/\gamma^* \to \tau\tau$, $t\bar{t}$ and $W+$jets are simulated with the same chain and mixed in proportions determined by their relative cross sections. Different aspects of the $Z/\gamma^* \to \tau\tau$ background are simulated with ALPGEN, PYTHIA8, Herwig, and SHERPA [66]; the details can be found in Table 1 of Ref. [58]. The $t\bar{t}$ background is simulated with POWHEG and PYTHIA8 and the $W+$jets background is simulated with ALPGEN [67] and PYTHIA8. Each event is characterized by 29 features,[2] including the lepton momenta and angles, the magnitude and direction of missing transverse momentum, the energy and angles of leading and subleading jets, and several other primary and derived variables. See Ref. [56] for details.

The most important nuisance parameter is the unknown absolute energy scale of the hadronically decaying τ leptons. We follow prior studies [52,59] and model this using a skewing function [69], which is applied to the τ lepton E_{T}, for signal and background alike. The minimum E_{T} threshold of 22 GeV is applied after skewing.

At the nominal value of the nuisance parameter, $z = 1$, the τ lepton energies are left unchanged. The impact of $z = 0.9$ or 1.1, on several features is shown in Fig. 7 for the background and Fig. 8 for the signal. While for the original ATLAS measurement the precision on this energy scale was $\pm(24)\%$ [58,70], this systematic uncertainty is deliberately inflated in our study due to the statistically limited dataset. The (unweighted) total number of events that pass the E_{T} threshold for the $z = 0.9$, $z = 1$ and $z = 1.1$ datasets are 618906, 719349, and 818201, respectively. The data are split into training and test set in the ratio 2∶1. Since the data at various values of z are generated from the nominal sample, the samples are to a large extent correlated. The train-test split therefore is determined before the skewing function and E_{T} threshold are applied, ensuring complete independence between training and test sets.

Thirty bins are used to construct the template and observed histograms.



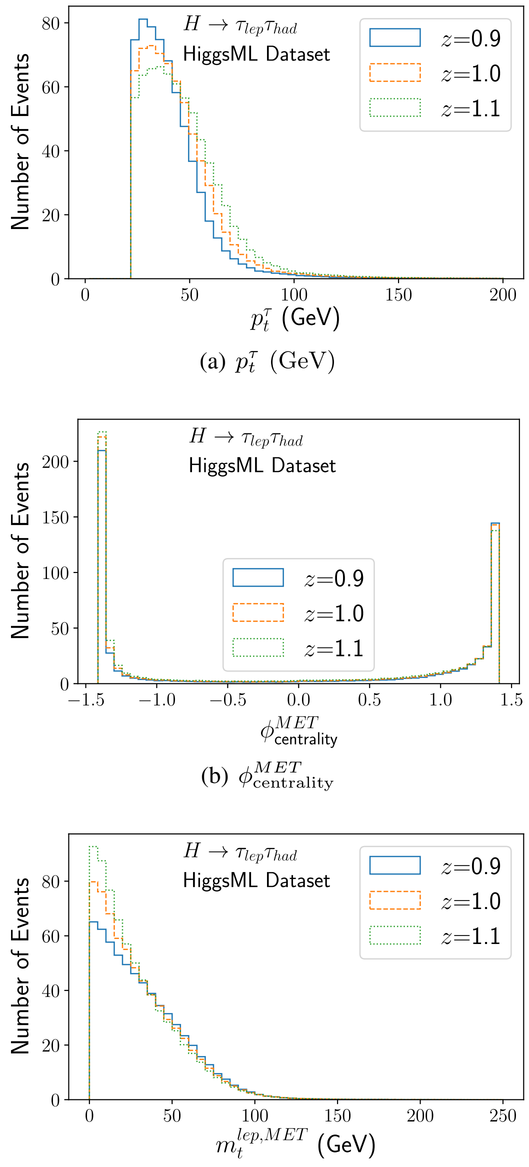
(a) p_t^τ (GeV)

(b) $\phi^{MET}_{\mathrm{centrality}}$

(c) $m_t^{lep,MET}$ (GeV)

FIG. 7. Distribution of physics variables for three values of the nuisance parameter that controls the absolute tau lepton energy scale, $z = \{0.9, 1, 1.1\}$ for background processes. (a) The transverse momentum of the hadronic τ, (b) the centrality in ϕ of the missing transverse energy vector with respect to the hadronic τ and the lepton, and (c) the transverse mass of the missing transverse energy and the lepton.

[2] The DER_mass_MMC feature listed in Ref. [56] was not included in the studies, following precedent set by Ref. [52], because the missing mass calculator [68] is slow to run and, as an Markov chain Monte Carlo algorithm, introduces an additional source of stochasticity, which makes comparisons difficult.



(a) p_t^{τ} (GeV)

(b) $\phi^{MET}_{\text{centrality}}$

(c) $m_t^{lep,MET}$ (GeV)

FIG. 8. Distribution of physics variables for three values of the nuisance parameter that controls the absolute tau lepton energy scale, $z = \{0.9, 1, 1.1\}$ for signal. (a) The transverse momentum of the hadronic τ, (b) the centrality in ϕ of the missing transverse energy vector with respect to the hadronic τ and the lepton, and (c) the transverse mass of the missing transverse energy and the lepton.

A. Description of trained models

All methods were implemented using neural networks. The baseline classifier was trained only on data at $z = 1$, while the data augmentation classifier, uncertainty-aware classifier and the adversarial classifier are all trained at 24 values spaced between $z = 0.7$ and $z = 1.4$. Two additional classifiers were also trained on data at $z = 0.8$ and $z = 1.1$ to estimate the best possible performance for an unparametrized classifier at these values of the nuisance parameter.

Technical details about the training procedure and architectures of the models are given below.

1. Baseline classifier

The neural network comprises 10 hidden layers with 512 nodes each, RLU activations and L2 kernel regularizers for all but the first hidden layer and a final layer with a single node and sigmoid activation. It was trained with an RMSProp optimizer, BCE loss and a batch size of 4096.

2. Data augmentation

The network comprises 10 hidden layers, each with 64 nodes, a RLU activation, and L2 kernel regularizers for all but the first hidden layer and a final layer with sigmoid activation. The network was trained with an Adam optimizer [71], BCE loss and a batch size of 4096.

3. Adversarial training

Both the classifier and the adversary consist of 10 hidden layers with 64 nodes and RLU activations and L2 kernel regularizers for all but the first hidden layer and a final layer with a single node with a sigmoid and linear activation for the classifier and adversary respectively. BCE is used as the classification loss and mean squared error as the adversarial loss. The two networks are attached with a gradient reversal layer which scales the gradient by -0.2 and trained with an RMSProp optimizer with $\lambda = 1$ and a batch size of 4096.

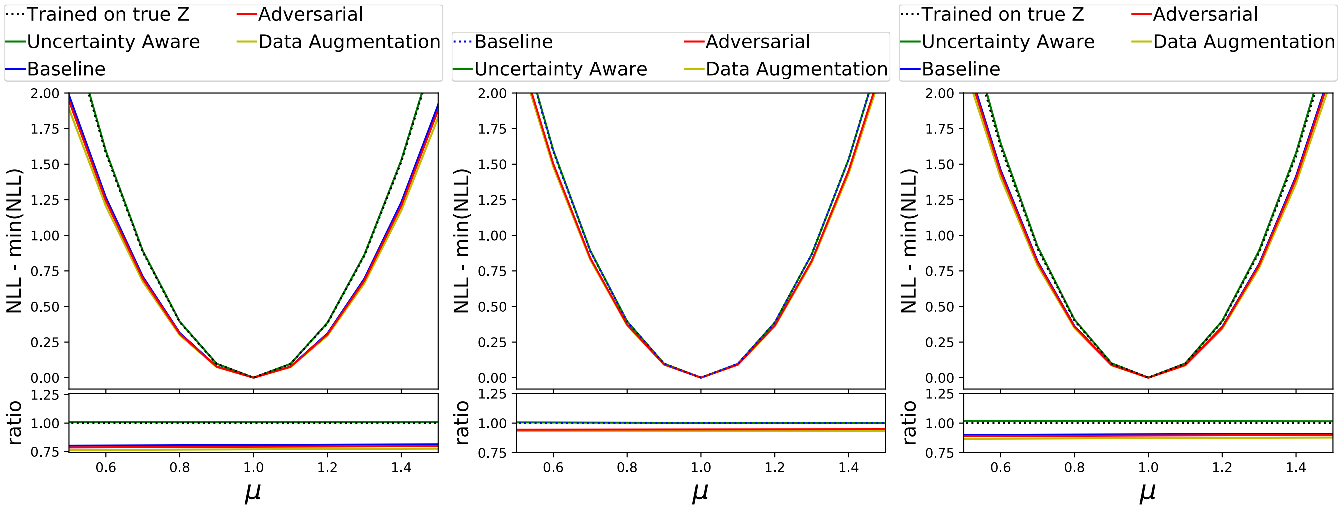
4. Uncertainty-aware classifier

The uncertainty-aware classifier is comprised of two subnetworks combined with a custom "if-else" layer that outputs the result of the first sub-network if the input z is less than 1 and the result of the second subnetwork otherwise. This approach allows training of one subnetwork longer than another if the performance in the $z \geq 1$ and $z < 1$ regions converge at different rates.

Each subnetwork consists of 10 hidden layers with 64 nodes, RLU activations and L2 kernel regularizers for all but the first hidden layer and a final layer with a single node and sigmoid activation. They were trained using RMSProp optimizer, BCE loss, and a batch size of 4096.

5. Classifiers trained on data from true z

Two neural networks were trained on data generated at $z = 0.8$ and $z = 1.1$, respectively. The network trained on data at $z = 0.8$ comprises 10 hidden layers with 64 nodes each, RLU activations and L2 kernel regularizers for all but the first hidden layer and a final layer with a single node with a sigmoid activation. The network trained on data at $z = 1.1$ has an identical structure except that the hidden layers are 512 nodes wide. Both networks were trained using an RMSProp optimizer, BCE loss, and a batch size of 4096.



(a) Systematic Down Data (b) Nominal Data (c) Systematic Up Data

FIG. 9. Physics dataset: profiled NLL curves for all four classifiers evaluated on (a) systematic down ($z_T = 0.8$), (b) nominal data ($z_T = 1.0$), and (c) systematic up data ($z_T = 1.1$) where the true value of μ is 1. Narrower curves indicate more precise measurements having accounted for systematic and statistical uncertainties.

B. Results

We evaluate the power of each method by examining the width of the profile likelihood curves in the parameter of interest, μ, see Fig. 9. The true value of μ was set to 1 (for comparisons when the true value of μ is set to 2 instead, refer to Appendix C). When the data are generated with $z = 1$, matching the assumption of the baseline classifier, we see that both the baseline and uncertainty-aware classifiers achieve the ideal performance, while the adversarial and data-augmented classifiers are slightly weaker. However, when the data are generated with a shift in the nuisance parameter ($z = 0.8, 1.1$) relative to the value used to build the baseline classifier, the uncertainty aware classifier maintains its ideal performance while the baseline classifier becomes less powerful.

VI. CONCLUSIONS

In this paper,[3] we have advocated for uncertainty-aware classifiers where the dependence on nuisance parameter is maximized during training by exploiting parametrized classifiers [29,30]. Using a Gaussian example and a realistic $H \to \tau\tau$ example, we have shown that the uncertainty-aware approach outperforms alternative methods that either are unaware of uncertainties or try to reduce the dependence on them during training. Our approach is successful because it provides the most effective classifier for all values of the nuisance parameter. This is useful when uncertainties are evaluated and when the nuisance parameter is profiled. It should be straightforward to apply this approach to multiple nuisance parameters although it was demonstrated on a single nuisance parameter in this paper.

While we advocate for maximally depending on the nuisance parameters, there could be cases where reducing the dependence could be beneficial. For example, eliminating the dependence on a particular nuisance parameter reduces the analysis complexity. If the classifier is used to make a simple selection (cut and count), then reducing the dependence could also improve analysis sensitivity when the uncertainty on the nuisance parameters is large [35]. For example, consider the case of an analysis that is considering adding a cut on a feature that is independent from all other features and that the cut has a signal efficiency ϵ_S, a background efficiency ϵ_B, and a relative uncertainty on ϵ_B that is δ. Clearly, one needs $\epsilon_S/\sqrt{\epsilon_B} \gtrsim 1$ for this cut to be useful. If δ is small, then the cut will certainly be useful. If δ is large, it may be beneficial to not make the cut on the feature, which is the analog of rendering the analysis insensitive to δ. Caution must be taken in such cases because reducing the sensitivity to a nuisance parameter may hide the size of the true uncertainty. This is the case for many two-point uncertainties such as hadronization modeling. Ultimately, the decision to use the additional feature or not depends on how the test statistic will be used in the analysis.

A related topic to uncertainty-awareness is inference awareness, where classifiers are trained using the final analysis objective as the loss. The ultimate sensitivity will be achieved when both of these approaches are combined, which will be the topic of future investigation.

Uncertainty awareness is a relatively straightforward extension of existing analyses performed at the LHC. Many nuisance parameters can be varied without significant computational overhead and with the minimal changes we

[3] The code for this paper can be found at https://github.com/hep-lbdl/systaware.

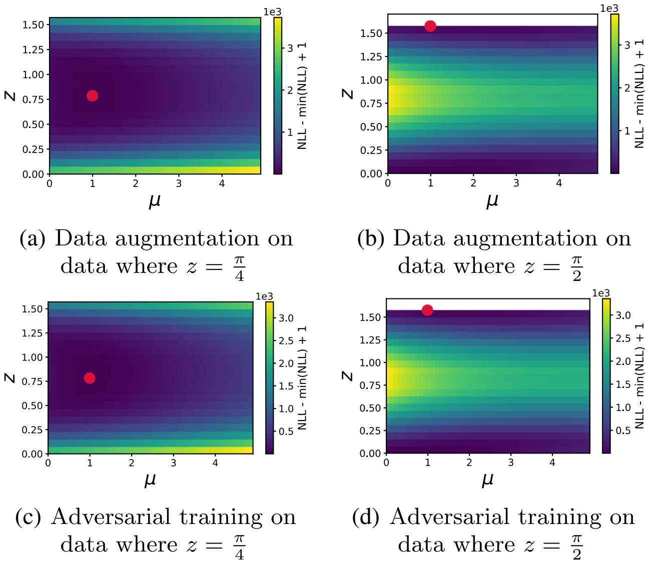
propose, analyses may be able to improve their sensitivity. The biggest improvements are expected for analyses that are limited by experimental systematic uncertainties. A growing number of analyses will fall into this category with the high-statistics of the high-luminosity LHC and other high energy physics experiments across frontiers.

ACKNOWLEDGMENTS

A. G. thanks David Rousseau and Victor Estrade for fruitful discussions over several years, and for sharing code to generate the physics dataset, and also thanks Glen Cowan for fruitful discussions. We thank Tommaso Dorigo, Victor Estrade, David Rousseau, and Kyle Cranmer for providing helpful comments on the manuscript. B. N. is supported by the U.S. Department of Energy (DOE), Office of Science under Contract No. DE-AC02-05CH11231. A. G. and D. W. are supported by the U.S. Department of Energy (DOE), Office of Science under Grant No. DE-SC0009920.

APPENDIX A: LIKELIHOOD SCANS AND ADDITIONAL COMPARISONS

Examples of the negative log-likelihood as a function of the parameter of interest μ and the nuisance parameter z are shown for data augmentation and adversarial training in Fig. 10 for the Gaussian example (Sec. IV). The same is shown for all four approaches compared for the realistic example (Sec. V) in Fig. 11.



(a) Data augmentation on data where $z = \frac{\pi}{4}$

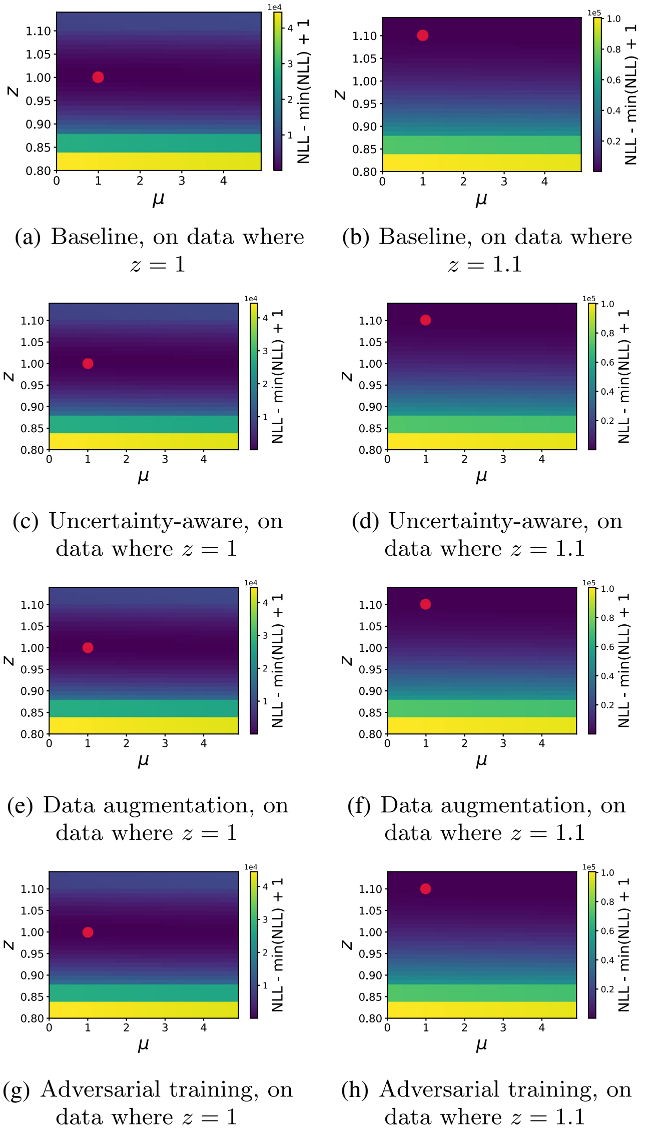
(b) Data augmentation on data where $z = \frac{\pi}{2}$

(c) Adversarial training on data where $z = \frac{\pi}{4}$

(d) Adversarial training on data where $z = \frac{\pi}{2}$

FIG. 10. The negative log-likelihood [Eq. (4)] as a function of the parameter of interest μ and the nuisance parameter z for two example datasets in the Gaussian example, using templates from the data augmentation (top) and adversarial training (bottom). On the left column, the data are generated with $z = \frac{\pi}{4}$, while in the right column, the data are generated with $z = \frac{\pi}{2}$. The red dot indicates the maximum likelihood estimate that coincides with the true value of μ, z in each case.

For the Gaussian example, a data augmented classifier is trained on an extended dataset with 5.4×10^7 events generated at 27 values of z for signal and background, extending the original range by three points at each extreme of the z range. It performs similarly to the original data-augmentation classifier. The adversarially trained classifier is studied in a higher decorrelation regime where an improved performance at $z = \frac{\pi}{2}$ comes at the cost of a deteriorated performance at $z = \frac{\pi}{4}$, as is to be expected.



(a) Baseline, on data where $z = 1$

(b) Baseline, on data where $z = 1.1$

(c) Uncertainty-aware, on data where $z = 1$

(d) Uncertainty-aware, on data where $z = 1.1$

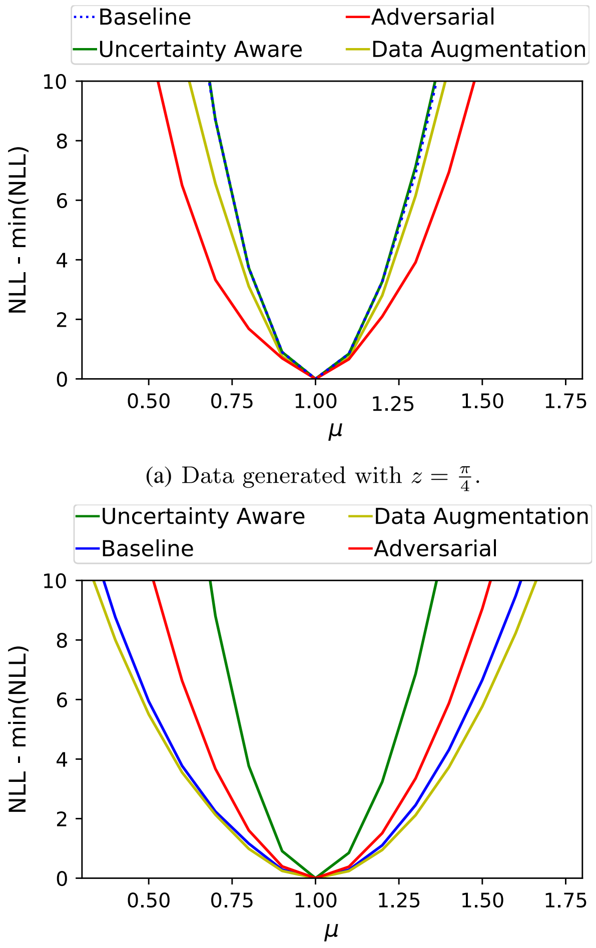
(e) Data augmentation, on data where $z = 1$

(f) Data augmentation, on data where $z = 1.1$

(g) Adversarial training, on data where $z = 1$

(h) Adversarial training, on data where $z = 1.1$

FIG. 11. The negative log-likelihood [Eq. (4)] as a function of the parameter of interest μ and the nuisance parameter z for two example datasets in the realistic example, using templates from the baseline (first row), systematic aware (second row), data augmentation (third row), and adversarial classifier (fourth row). On the left column, the data are generated with $z = 1$, while on the right column, the data are generated with $z = 1.1$. The red dot indicates the maximum likelihood estimate which coincides with the true value of μ, z in each case. Note that the z-axis scale is not uniform in all figures.



(a) Data generated with $z = \frac{\pi}{4}$.

(b) Data generated with $z = \frac{\pi}{2}$.

FIG. 12. The profile likelihood $\max_z \mathcal{L}(\mu, z)$ as a function of the parameter of interest, μ for likelihoods calculated with templates built from the updated data-augmentation and adversarially trained classifiers in comparison to the baseline and uncertainty-aware classifiers. This adversarially trained classifier performs significantly worse for data generated with $z = \frac{\pi}{4}$ (top) but performs significantly better for data generated at $z = \frac{\pi}{2}$ compared to the one discussed in Sec. IV. The data-augmentation classifier performs similarly to the one discussed in Sec. IV.

To improve the stability of training for an increased the level of invariance, the network architecture is updated. A $\lambda = 10^4$ is used, the gradient reversal layer scales the gradients by -1 and a mixture density network [72] with two Gaussian components is used for the regression task of the adversary. It comprises two hidden layers with 500 nodes and RLU activations. The output layer consists of six nodes, two each for the mean, standard deviation and the mixing coefficient for the two Gaussian components. The activations for the mean nodes are linear for the standard deviation nodes are non-negative exponential linear units (exponential linear units offset by 1) and for the mixing coefficient nodes are Softmax. Figure 12 shows the performance of these two classifiers as compared to the baseline and uncertainty-aware classifiers.

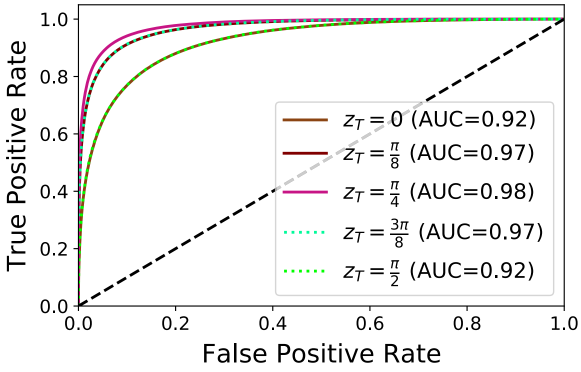


FIG. 13. ROC curves for the baseline classifier evaluated on datasets generated at various values of z. A larger area under the curve (AUC) indicates better separation power. Since the baseline classifier assumes $z = \frac{\pi}{4}$, the separation power between signal and background is maximum for data generated at $z = \frac{\pi}{4}$ and deteriorates for data generated at z values further away in either direction.

The performance of the baseline classifier for datasets from a few values of z is shown in Fig. 13. Since the baseline classifier assumes $z = \frac{\pi}{4}$, it performs best when evaluated on data with $z = \frac{\pi}{4}$ and the classification power deteriorates for other values of z.

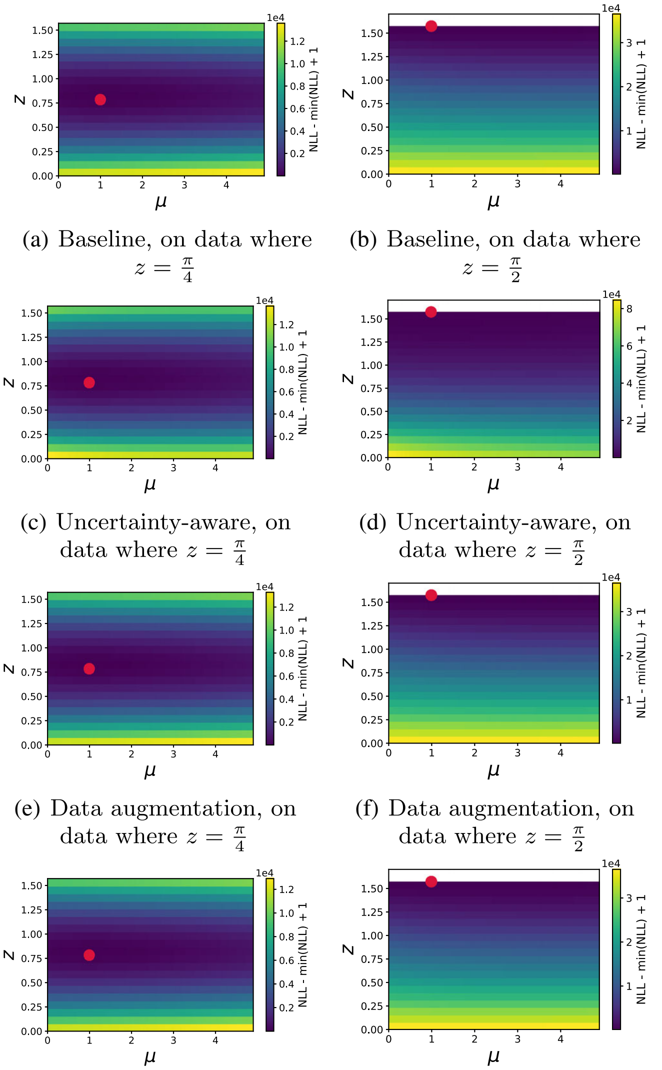
APPENDIX B: GAUSSIAN EXAMPLE WITH AUXILIARY MEASUREMENT OF z

A study is performed by replacing the prior on z in Eq. (4) with a simultaneous auxiliary measurement. For simplicity the auxiliary measurement is of a Gaussian distribution with mean at z_{T} and standard deviation of 0.5. 10^5 events uniformly weighted 0.1 are generated at each of the 21 values of z. The negative log-likelihood then reads,

$$\begin{aligned} &-\log \mathcal{L}(\mu, z|\{x_i\}) \\ &\quad = -\sum_{j=1}^{n_{\mathrm{bins}}} [N_j \cdot \log(\mu s_j + b_j) - \mu s_j - b_j - \log(\Gamma(N_i))] \\ &\qquad - \sum_{k=1}^{m_{\mathrm{bins}}} [N_k^{\mathrm{aux}} \cdot \log(a_k^z) - a_k^z - \log(\Gamma(N_k^{\mathrm{aux}})], \end{aligned} \tag{B1}$$

where a_k^z is the number of events expected in bin k of the auxiliary measurement for $z_{\mathrm{T}} = z$ and N_k^{aux} is the number of events actually observed in that bin. Four bins are used to construct the template and observed histograms for the auxiliary measurement.

The classifiers described in Sec. IV are reused for this study, no retraining is required. The likelihood scans for the various approaches are shown in Fig. 14. For data



(a) Baseline, on data where $z = \frac{\pi}{4}$

(b) Baseline, on data where $z = \frac{\pi}{2}$

(c) Uncertainty-aware, on data where $z = \frac{\pi}{4}$

(d) Uncertainty-aware, on data where $z = \frac{\pi}{2}$

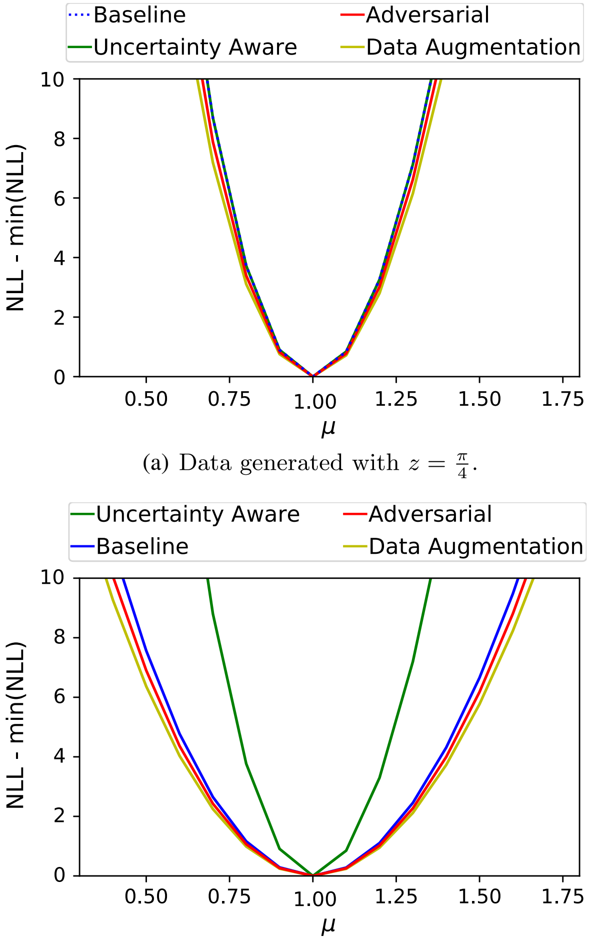
(e) Data augmentation, on data where $z = \frac{\pi}{4}$

(f) Data augmentation, on data where $z = \frac{\pi}{2}$

(g) Adversarial training, on data where $z = \frac{\pi}{4}$

(h) Adversarial training, on data where $z = \frac{\pi}{2}$

FIG. 14. The negative log-likelihood [Eq. (B1)] as a function of the parameter of interest μ and the nuisance parameter z in the auxiliary measurement study, using templates from the baseline (first row), systematic aware (second row), data augmentation (third row), and adversarial classifier (fourth row). On the left column, the data are generated with $z = \frac{\pi}{4}$, while on the right column, the data are generated with $z = \frac{\pi}{2}$. The red dot indicates the maximum likelihood estimate that coincides with the true value of μ, z in each case. Note that the z-axis scale is not uniform in all figures.



(a) Data generated with $z = \frac{\pi}{4}$.

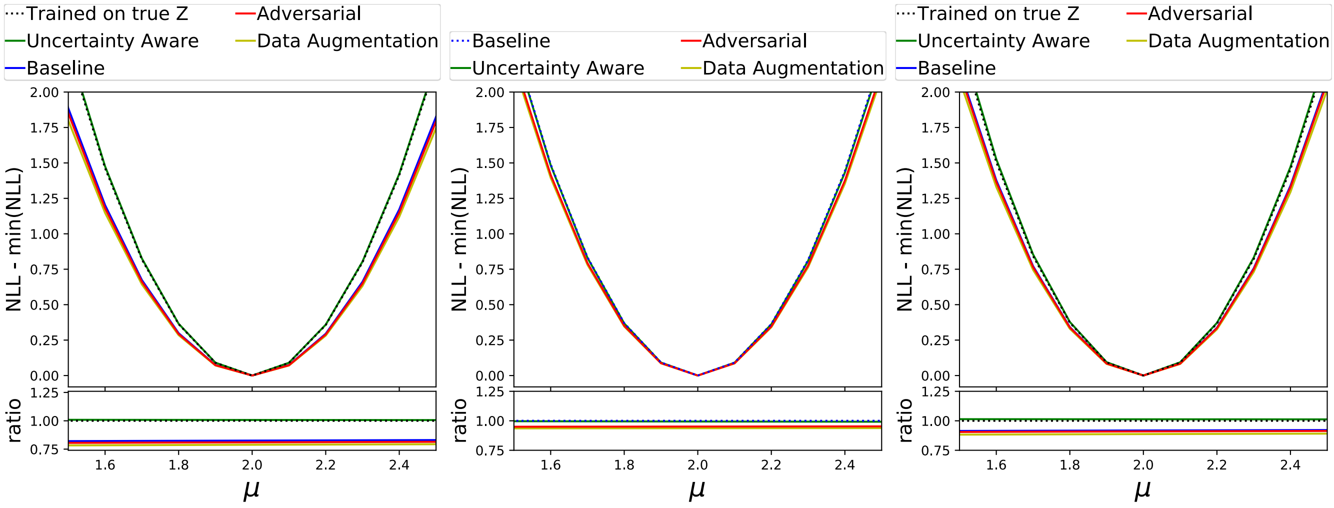
(b) Data generated with $z = \frac{\pi}{2}$.

FIG. 15. The profile likelihood $\max_z \mathcal{L}(\mu, z)$ as a function of the parameter of interest, μ for likelihoods calculated with templates built from the various classifiers in the auxiliary measurement study. The baseline classifier assumes $z = \frac{\pi}{4}$, and matches the performance of the uncertainty-aware classifier in data generated with $z = \frac{\pi}{4}$ (top). In data generated with $z = \frac{\pi}{2}$, the power of all classifiers other than the uncertainty-aware classifier become significantly weaker despite a better constraint on z compared to Sec. IV.

generated at $z = \frac{\pi}{2}$ all approaches can exclude $z = 0$ since the auxiliary measurement constrains z much more than the prior used in Sec. IV, Fig. 5. The profile likelihood in Fig. 15 shows that although the curves are narrower compared to Fig. 6, the overall conclusions discussed in Sec. IV remain valid.

APPENDIX C: TESTS AT $\mu = 2$ FOR PHYSICS EXAMPLE

The comparison of the four approaches was also performed for data where the true value of the parameter of interest μ is 2. The profile likelihoods in Fig. 16 show that the conclusions of Sec. V remain valid.



(a) Systematic Down Data

(b) Nominal Data

(c) Systematic Up Data

FIG. 16. Physics dataset: profiled NLL curves for all four classifiers evaluated on (a) systematic down ($z_{\mathrm{T}} = 0.8$), (b) nominal data ($z_{\mathrm{T}} = 1.0$), and (c) systematic up data ($z_{\mathrm{T}} = 1.1$) where the true value of μ is 2. Narrower curves indicate more precise measurements having accounted for systematic and statistical uncertainties.

[1] G. Aad *et al.* (ATLAS, CMS Collaborations), Combined Measurement of the Higgs Boson Mass in pp Collisions at $\sqrt{s} = 7$ and 8 TeV with the ATLAS and CMS Experiments, Phys. Rev. Lett. **114**, 191803 (2015).

[2] W. L. Freedman and B. F. Madore, The hubble constant, Annu. Rev. Astron. Astrophys. **48**, 673 (2010).

[3] A. J. Larkoski, I. Moult, and B. Nachman, Jet substructure at the large hadron collider: A review of recent advances in theory and machine learning, Phys. Rep. **841**, 1 (2020).

[4] D. Guest, K. Cranmer, and D. Whiteson, Deep learning and its application to LHC physics, Annu. Rev. Nucl. Part. Sci. **68**, 161 (2018).

[5] K. Albertsson *et al.*, Machine learning in high energy physics community white paper, J. Phys. Conf. Ser. **1085**, 022008 (2018).

[6] A. Radovic, M. Williams, D. Rousseau, M. Kagan, D. Bonacorsi, A. Himmel, A. Aurisano, K. Terao, and T. Wongjirad, Machine learning at the energy and intensity frontiers of particle physics, Nature (London) **560**, 41 (2018).

[7] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborov, Machine learning and the physical sciences, Rev. Mod. Phys. **91**, 045002 (2019).

[8] D. Bourilkov, Machine and deep learning applications in particle physics, Int. J. Mod. Phys. A **34**, 1930019 (2019).

[9] M. D. Schwartz, Modern machine learning and particle physics, arXiv:2103.12226.

[10] A. Blance, M. Spannowsky, and P. Waite, Adversarially-trained autoencoders for robust unsupervised new physics searches, J. High Energy Phys. 10 (2019) 047.

[11] C. Englert, P. Galler, P. Harris, and M. Spannowsky, Machine learning uncertainties with adversarial neural networks, Eur. Phys. J. C **79**, 4 (2019).

[12] G. Louppe, M. Kagan, and K. Cranmer, Learning to pivot with adversarial networks, arXiv:1611.01046.

[13] J. Dolen, P. Harris, S. Marzani, S. Rappoccio, and N. Tran, Thinking outside the ROCs: Designing Decorrelated Taggers (DDT) for jet substructure, J. High Energy Phys. 05 (2016) 156.

[14] I. Moult, B. Nachman, and D. Neill, Convolved substructure: Analytically decorrelating jet substructure observables, J. High Energy Phys. 05 (2018) 002.

[15] J. Stevens and M. Williams, uBoost: A boosting method for producing uniform selection efficiencies from multivariate classifiers, J. Instrum. **8**, P12013 (2013).

[16] C. Shimmin, P. Sadowski, P. Baldi, E. Weik, D. Whiteson, E. Goul, and A. Sgaard, Decorrelated jet substructure tagging using adversarial neural networks, Phys. Rev. D **96**, 074034 (2017).

[17] L. Bradshaw, R. K. Mishra, A. Mitridate, and B. Ostdiek, Mass agnostic jet taggers, SciPost Phys. **8**, 011 (2020).

[18] The ATLAS Collaboration, Performance of mass-decorrelated jet substructure observables for hadronic two-body decay tagging in ATLAS, Report No. ATL-PHYS-PUB-2018-014, 2018.

[19] G. Kasieczka and D. Shih, DisCo Fever: Robust Networks Through Distance Correlation, Phys. Rev. Lett. **125**, 122001 (2020).

[20] S. Wunsch, S. Jórger, R. Wolf, and G. Quast, Reducing the dependence of the neural network function to systematic

uncertainties in the input space, Comput. Software Big Sci. **4**, 5 (2020).

[21] A. Rogozhnikov, A. Bukva, V. V. Gligorov, A. Ustyuzhanin, and M. Williams, New approaches for boosting to uniformity, J. Instrum. **10**, T03002 (2015).

[22] CMS Collaboration, A deep neural network to search for new long-lived particles decaying to jets, Mach. Learn.: Sci. Technol. **1**, 035012 (2020).

[23] J. M. Clavijo, P. Glaysher, and J. M. Katzy, Adversarial domain adaptation to reduce sample bias of a high energy physics classifier, arXiv:2005.00568.

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

[25] O. Kitouni, B. Nachman, C. Weisser, and M. Williams, Enhancing searches for resonances with machine learning and moment decomposition, J. High Energy Phys. 04 (2020) 070.

[26] V. Estrade, C. Germain, I. Guyon, and D. Rousseau, Systematic aware learning—A case study in high energy physics, EPJ Web Conf. **214**, 06024 (2019).

[27] T. Sjostrand, S. Mrenna, and P. Z. Skands, A brief introduction to PYTHIA8.1, Comput. Phys. Commun. **178**, 852 (2008).

[28] J. Bellm *et al.*, Herwig 7.0/Herwig++ 3.0 release note, Eur. Phys. J. C **76**, 196 (2016).

[29] K. Cranmer, J. Pavez, and G. Louppe, Approximating likelihood ratios with calibrated discriminative classifiers, arXiv:1506.02169.

[30] P. Baldi, K. Cranmer, T. Faucett, P. Sadowski, and D. Whiteson, Parameterized neural networks for high-energy physics, Eur. Phys. J. C **76**, 235 (2016).

[31] J. Brehmer, F. Kling, I. Espejo, and K. Cranmer, MadMiner: Machine learning-based inference for particle physics, Comput. Software Big Sci. **4**, 3 (2020).

[32] J. Brehmer, G. Louppe, J. Pavez, and K. Cranmer, Mining gold from implicit models to improve likelihood-free inference, Proc. Natl. Acad. Sci. U.S.A. **117**, 5242 (2020).

[33] J. Brehmer, K. Cranmer, G. Louppe, and J. Pavez, Constraining Effective Field Theories with Machine Learning, Phys. Rev. Lett. **121**, 111801 (2018).

[34] J. Brehmer, K. Cranmer, G. Louppe, and J. Pavez, A guide to constraining effective field theories with machine learning, Phys. Rev. D **98**, 052004 (2018).

[35] B. Nachman, A guide for deploying Deep Learning in LHC searches: How to achieve optimality and account for uncertainty, SciPost Phys. **8**, 090 (2020).

[36] S. Wunsch, S. Jrger, R. Wolf, and G. Quast, Optimal statistical inference in the presence of systematic uncertainties using neural network optimization based on binned Poisson likelihoods with nuisance parameters, Comput Software Big. Sci. **5**, 4 (2021).

[37] A. Elwood, D. Krücker, and 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).

[38] L.-G. Xia, QBDT, a new boosting decision tree method with systematical uncertainties into training for High Energy Physics, Nucl. Instrum. Methods Phys. Res., Sect. A **930**, 15 (2019).

[39] P. De Castro and T. Dorigo, INFERNO: Inference-aware neural optimisation, Comput. Phys. Commun. **244**, 170 (2019).

[40] T. Charnock, G. Lavaux, and B. D. Wandelt, Automatic physical inference with information maximizing neural networks, Phys. Rev. D **97**, 083004 (2018).

[41] J. Alsing and B. Wandelt, Nuisance hardened data compression for fast likelihood-free inference, Mon. Not. R. Astron. Soc. **488**, 5093 (2019).

[42] L. Heinrich and N. Simpson, pyhf/neos: initial zenodo release, https://doi.org/10.5281/zenodo.3697981 (2020).

[43] G. Kasieczka, M. Luchmann, F. Otterpohl, and T. Plehn, Per-object systematics using deep-learned calibration, SciPost Phys. **9**, 089 (2020).

[44] S. Bollweg, M. Haumann, G. Kasieczka, M. Luchmann, T. Plehn, and J. Thompson, Deep-learning jets with uncertainties and more, SciPost Phys. **8**, 006 (2020).

[45] J. Y. Araz and M. Spannowsky, Combine and conquer: Event reconstruction with bayesian ensemble neural networks, J. High Energy Phys. 04 (2021) 296.

[46] M. Bellagente, M. Haußmann, M. Luchmann, and T. Plehn, Understanding event-generation networks via uncertainties, arXiv:2104.04543.

[47] T. Dorigo and P. de Castro, Dealing with nuisance parameters using machine learning in high energy physics: A review, arXiv:2007.09121.

[48] F. Chollet *et al.*, Keras (2015), https://keras.io.

[49] M. Abadi *et al.*, TensorFlow: Large-scale machine learning on heterogeneous systems (2015), software available from tensorflow.org.

[50] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer Series in Statistics (Springer New York Inc., New York, NY, USA, 2001).

[51] Y. Ganin and V. Lempitsky, Unsupervised domain adaptation by backpropagation, arXiv:1409.7495.

[52] V. Estrade, C. Germain, I. Guyon, and D. Rousseau, Adversarial learning to eliminate systematic errors: A case study in High Energy Physics, EPJ Web Conf. **214**, 06024 (2017).

[53] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, Scikit-learn: Machine learning in PYTHON, J. Mach. Learn. Res. **12**, 2825 (2011).

[54] C. Cortes, M. Mohri, and A. Rostamizadeh, L2 regularization for learning kernels, arXiv:1205.2653.

[55] T. Tieleman and G. Hinton, Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude, *COURSERA: Neural Networks for Machine Learning* (2012).

[56] ATLAS Collaboration, Dataset from the ATLAS Higgs Boson Machine Learning Challenge 2014. CERN Open Data Portal.

[57] C. Adam-Bourdarios, G. Cowan, C. Germain, I. Guyon, B. Kgl, and D. Rousseau, The Higgs boson machine learning challenge, in *Proceedings of the NIPS 2014 Workshop on High-energy Physics and Machine Learning*, edited by G. Cowan, C. Germain, I. Guyon, B. Kgl, and D. Rousseau (PMLR, Montreal, Canada, 2015), Vol. 42, pp. 19–55.

[58] G. Aad *et al.* (ATLAS Collaboration), Evidence for the Higgs-boson Yukawa coupling to tau leptons with the ATLAS detector, J. High Energy Phys. 04 (2015) 117.

[59] V. Estrade, C. Germain, I. Guyon, and D. Rousseau, Systematic aware learning—a case study in high energy physics, EPJ Web Conf. **214**, 06024 (2019).

[60] P. Nason, A new method for combining NLO QCD with shower Monte Carlo algorithms, J. High Energy Phys. 11 (2004) 040.

[61] S. Frixione, P. Nason, and C. Oleari, Matching NLO QCD computations with parton shower simulations: The POWHEG method, J. High Energy Phys. 11 (2007) 070.

[62] S. Alioli, P. Nason, C. Oleari, and E. Re, A general framework for implementing NLO calculations in shower Monte Carlo programs: The POWHEG BOX, J. High Energy Phys. 06 (2010) 043.

[63] E. Bagnaschi, G. Degrassi, P. Slavich, and A. Vicini, Higgs production via gluon fusion in the POWHEG approach in the SM and in the MSSM, J. High Energy Phys. 02 (2012) 088.

[64] S. Agostinelli *et al.* (GEANT4 Collaboration), GEANT4–a simulation toolkit, Nucl. Instrum. Methods Phys. Res., Sect. A **506**, 250 (2003).

[65] ATLAS Collaboration, Athena, https://doi.org/10.5281/zenodo.2641997(2019).

[66] T. Gleisberg, S. Hoeche, F. Krauss, M. Schonherr, S. Schumann, F. Siegert, and J. Winter, Event generation with SHERPA1.1, J. High Energy Phys. 02 (2009) 007.

[67] M. L. Mangano, M. Moretti, F. Piccinini, R. Pittau, and A. D. Polosa, ALPGEN, a generator for hard multiparton processes in hadronic collisions, J. High Energy Phys. 07 (2003) 001.

[68] A. Elagin, P. Murat, A. Pranko, and A. Safonov, A new mass reconstruction technique for resonances decaying to di-tau, Nucl. Instrum. Methods Phys. Res., Sect. A **654**, 481 (2011).

[69] V. Estrade, victor-estrade/datawarehouse: First release, https://doi.org/10.5281/zenodo.1887847 (2018).

[70] ATLAS Collaboration, Determination of the tau energy scale and the associated systematic uncertainty in proton-proton collisions at $\sqrt{s} = 8$ TeV with the ATLAS detector at the LHC in 2012 (2013), https://cds.cern.ch/record/1544036.

[71] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv:1412.6980.

[72] C. M. Bishop, Mixture density networks (1994).