Bias and priors in machine learning calibrations for high energy physics

Rikab Gambhir,[1,2,*] Benjamin Nachman,[3,4,†] and Jesse Thaler[1,2,‡]

[1]*Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA*

[2]*The NSF AI Institute for Artificial Intelligence and Fundamental Interactions*

[3]*Physics Division, Lawrence Berkeley National Laboratory, Berkeley, California 94720, USA*

[4]*Berkeley Institute for Data Science, University of California, Berkeley, California 94720, USA*

(Received 23 May 2022; accepted 20 July 2022; published 15 August 2022)

Machine learning offers an exciting opportunity to improve the calibration of nearly all reconstructed objects in high-energy physics detectors. However, machine learning approaches often depend on the spectra of examples used during training, an issue known as prior dependence. This is an undesirable property of a calibration, which needs to be applicable in a variety of environments. The purpose of this paper is to explicitly highlight the prior dependence of some machine-learning-based calibration strategies. We demonstrate how some recent proposals for both simulation-based and data-based calibrations inherit properties of the sample used for training, which can result in biases for downstream analyses. In the case of simulation-based calibration, we argue that our recently proposed Gaussian Ansatz approach can avoid some of the pitfalls of prior dependence, whereas prior-independent data-based calibration remains an open problem.

DOI: 10.1103/PhysRevD.106.036011

I. INTRODUCTION

Calibration is the task of removing bias from an inference—that is, to ensure the inference is "correct on average." There are two major classes of calibration: simulation-based calibration, where the goal is to infer a truth reference object, and data-based calibration, where the goal is to match simulation and data distributions.

Both simulation-based calibrations and data-based calibrations are essential components of the experimental program in high-energy physics (HEP), and a significant amount of time is spent deriving these results to enable downstream analyses. We focus on the ATLAS and CMS experiments at the Large Hadron Collider (LHC) for our examples, but this discussion is relevant for all of HEP (and really any experiment). ATLAS and CMS have performed many recent calibrations, including the energy calibration of single hadrons [1,2], jets [3,4], muons [5,6], electrons/photons [7–9], and τ leptons [10,11]. The reconstruction efficiencies of all of these objects are also calibrated and include the classification efficiency of jets from heavy flavor [12,13] and even more massive particles [14,15].

Machine learning is a promising tool to improve both types of calibration. In particular, machine learning methods can readily process high-dimensional inputs and therefore can incorporate more information to improve the precision and accuracy of a calibration. There have been a large number of proposals for improving the simulation-based calibrations of various object energies, including single hadrons [16–21], muons [22], and jets [23–33] at colliders; kinematic reconstruction in deep inelastic scattering [34]; and neutrino energies in a variety of experiments [35–40]. Further ideas can be found in Ref. [41]. For data-based calibration, a machine learning procedure was recently proposed in Ref. [42].

Caution is needed to ensure that calibrations resulting from a machine learning approach satisfy certain important properties. One critical property of a calibration is that it should be *universal*—a calibration derived in one place should be applicable elsewhere. A nonuniversal calibration would have a rather limited utility and can produce undesirable results if applied to a dataset that does not exactly match the calibration dataset. Statistically, universality is synonymous with prior independence. Most of the existing machine-learning-based calibration proposals, though, are inherently prior dependent, as we will explain below.

A second critical property of a calibration is *closure*, which means that on average, the calibration produces the

*rikab@mit.edu
†bpnachman@lbl.gov
‡jthaler@mit.edu

correct answer.[1] To quantify closure, one often computes the *bias* of a calibration, which is the average deviation of the calibrated result from the target value. A calibration can be biased due to the choice of estimator or fitting procedure used, even if the usual pitfalls of dataset-induced biases are taken care of. As explained below, universality and closure are related, and a prior-dependent calibration will necessarily have irreducible bias.[2]

In this paper, we explain the origin of prior dependence for common calibration techniques, with explicit illustrative examples, and demonstrate the associated bias that these procedures incur. For simulation-based calibrations, we advocate for our Gaussian Ansatz [43] as a machine-learning-based strategy that is prior independent and bias-free. For data-based calibrations, we are unaware of any prior-independent methods in the literature. We hope that by highlighting these issues, we can inspire the development of prior-independent calibration methods.

The remainder of this paper is organized as follows. In Sec. II, we review the statistical properties of machine-learning-based calibration. In Sec. III, we clarify the meaning of *resolution* and *uncertainty* in the HEP context. To demonstrate the issue of prior dependence, we present Gaussian examples in Sec. IV. In Sec. V, we study an HEP application of calibration in the context of jet energy measurements at the LHC. The paper ends in Sec. VI with our conclusions and outlook.

II. THE STATISTICS OF CALIBRATION

In this section, we review some of the basic features of simulated-based and data-based calibration, and discuss the issues of prior dependence and bias.

A. Simulation-based calibration

In simulation-based calibration, the goal is to infer target (or true) features $z_T \in \mathbb{R}^N$ from detector-level features $x_D \in \mathbb{R}^M$—that is, to construct an *estimator* or *calibration function* $f\colon \mathbb{R}^M \to \mathbb{R}^N$ where

$$\hat{z}_T = f(x_D) \tag{1}$$

is the inferred estimate. To carry out simulation-based calibration, one starts with a set of (x_D, z_T) pairs, which typically come from an in-depth numerical simulation of an experiment. For the case study in Sec. V, x_D will be the experimentally measurable features of hadronic jets and z_T will be the true jet energy.

[1]Any measure of central tendency can be used to measure closure, such as the median or mode. In this paper, we will focus on the mean, as it is the usual target in machine learning and HEP applications.

[2]Prior independence is a necessary prerequisite for closure. However, even with prior independence, closure is not guaranteed.

For concreteness, one can think of the calibration function f as being parametrized by a universal function approximator such as a neural network, whose weights and biases are learned. This is often done by minimizing the mean squared error (MSE) loss:

$$f_{\mathrm{MSE}} = \underset{g}{\mathrm{argmin}} \mathbb{E}_{\mathrm{train}}[(g(X_D) - Z_T)^2], \tag{2}$$

where capital letters correspond to random variables and $\mathbb{E}$ represents the expectation value over the *training sample* used to derive the calibration. The calibration function is then deployed on the *testing sample*, which could be the dataset of interest or a hold-out control region.

Using the calculus of variations, one can show that with enough training data, a flexible enough functional parametrization, and a sufficiently exhaustive training procedure, the asymptotic solution to Eq. (2) is

$$f_{\mathrm{MSE}}(x_D) = \mathbb{E}_{\mathrm{train}}[Z_T | X_D = x_D], \tag{3}$$

where lowercase letters correspond to an instance of a random variable. In this way, f learns the mean value of z_T for a given x_D in the training set. Alternative loss functions result in statistics other than the mean. See, e.g., Ref. [44] for alternative approaches, including mode learning, which is a standard target for many traditional calibrations (usually in the form of truncated Gaussian fits; see, e.g., [9]).

B. Prior dependence and bias

A key assumption of simulation-based calibration is that the detector response is universal:

$$p_{\mathrm{test}}(x_D|z_T) = p_{\mathrm{train}}(x_D|z_T). \tag{4}$$

This equation says that for a given truth input z_T, the detector response is the same between the training data used for deriving the calibration and the testing data used for deploying the calibration. In some cases, the detector response might depend on more features than z_T, and if these hidden features are mismodeled, then Eq. (4) may not hold. For our analysis of simulation-based calibration, we assume Eq. (4) throughout.

Calibrations of the form of Eq. (3) are *not* universal, even if the detector response is. Writing out the MSE-based calibration in integral form, we have

$$\begin{aligned} f_{\mathrm{MSE}}(x_D) &= \int dz_T\, z_T p_{\mathrm{train}}(z_T|x_D) \\ &= \int dz_T\, z_T p_{\mathrm{train}}(x_D|z_T) \frac{p_{\mathrm{train}}(z_T)}{p_{\mathrm{train}}(x_D)}. \end{aligned} \tag{5}$$

Here, we have used Bayes' theorem to make explicit the dependence of f on $p_{\mathrm{train}}(z_T)$, the prior of true values used

for the training. Thus, even if $p_{\text{train}}(x_D|z_T)$ is universal via Eq. (4), the truth distribution is not:

$$p_{\text{test}}(z_T) \neq p_{\text{train}}(z_T). \tag{6}$$

The nonuniversality of the calibration function leads to bias, as we now explain.

The bias $b(z_T)$ of a calibration quantifies the degree of nonclosure. Specifically, bias is the average difference between the reconstructed value and the truth reference value. It is evaluated over the test sample, conditioned on the truth values:

$$b(z_T) = \mathbb{E}_{\text{test}}[f(X_D) - z_T|Z_T = z_T]. \tag{7}$$

A bias of zero means that, on average, the reconstructed and truth values agree. For MSE regression, the bias is

$$\begin{aligned} b(z_T) + z_T &= \int dx_D f_{\text{MSE}}(x_D) p_{\text{test}}(x_D|z_T) \\ &= \int dx_D dz'_T z'_T p_{\text{train}}(z'_T|x_D) p_{\text{test}}(x_D|z_T). \end{aligned} \tag{8}$$

This bias is dependent on the training prior through $p_{\text{train}}(z'_T|x_D)$. Thus, a prior-dependent calibration is *necessarily* biased, since it depends on the choice of $p_{\text{train}}(z_T)$.[3] Note that even if the training dataset is statistically identical to the testing dataset [i.e., $p_{\text{test}}(x_D, z_T) = p_{\text{train}}(x_D, z_T)$], it is not guaranteed that the calibration will be unbiased.

One way to reduce the bias is if the prior is "wide and flat enough," such that the prior asymptotically approaches a uniform sampling over the real line relative to the detector response. For example, one can show using Eq. (8) that if the prior $p(z_T)$ is Gaussian with standard deviation σ, the detector response $p(x_D|z_T)$ is a Gaussian noise model with standard deviation ϵ, and the test set is statistically identical to the training set, then the bias scales as

$$b(z_T) \sim \left(\frac{\epsilon}{\sigma}\right)^2 z_T + \mathcal{O}\left(\left(\frac{\epsilon}{\sigma}\right)^4\right). \tag{9}$$

In cases with steeply falling spectra, as is common in HEP, prior dependence usually leads to large biases in calibration, even if the testing and training sets follow the same distribution.

[3]Note that the bias does *not* depend on the choice of testing prior, $p_{\text{test}}(z_T)$, but rather only on $p_{\text{test}}(x_D|z_T)$. Depending on the choice of $p_{\text{test}}(x_D|z_T)$, it is possible for the bias to be zero, but this does not imply the inference is prior independent. For example, if $p_{\text{test}}(x_D|z_T) = \delta(x_D - z_T)$, and $\mathbb{E}_{\text{train}}[x_D|Z_T = z_T] = z_T$, then one can show that $b(z_T) = 0$.

C. Mitigating prior dependence

A majority of simulation-based calibrations (with or without machine learning) are set up using the MSE loss as described above, which means that they are biased. That said, there are alternative methods to mitigate the prior dependence and thereby reduce the bias. For example, simulation-based jet calibrations at the LHC use a technique called *numerical inversion* (see, e.g., Ref. [45]). The idea of numerical inversion is to regress x_D from z_T with a function $g(z_T)$ and then define the calibration function through the inverse:

$$f_{\text{NI}}(x_D) = g^{-1}(x_D). \tag{10}$$

Traditionally, x_D is one dimensional and g is parametrized with functions that can easily be inverted numerically, hence the name. The function g is given by

$$g(z_T) = \mathbb{E}_{\text{train}}[X_D|Z_T = z_T]. \tag{11}$$

Since the detector response $p(x_D|z_T)$ is universal, g is universal, and thus the derived f is also universal. Under certain assumptions, the f from numerical inversion is also unbiased [45].

Numerical inversion has been extended to work with neural networks [23,24], where the inversion step is accomplished with a second neural network. Alternatively, it may be possible to also achieve this with a natively invertible neural network such as a normalizing flow [46,47]. A key challenge with numerical inversion and its neural network generalizations are that they do not scale well to high dimensions.

In Ref. [43], we propose an alternative way to achieve a prior-independent calibration that scales well to high- and variable-dimensional settings. This approach is based on finding the local maximum likelihood, such that the learned calibration function becomes

$$f_{\text{MLC}}(x_D) = \underset{z_T}{\text{argmax}}\, p_{\text{train}}(x_D|z_T), \tag{12}$$

where MLC stands for maximum likelihood classifier—see Ref. [48]. Again, because the detector response $p(x_D|z_T)$ is universal, maximum likelihood calibrations are universal[4] and, in certain configurations, are provably unbiased. In particular, if the detector response $p(x_D|z_T)$ is a Gaussian noise model centered on z_T, then one can show that the bias is zero using Eq. (7):

[4]One important caveat is that universality here means prior independence over the space of priors that share the same support as the training set. One cannot get away with training a model on a single z_T instance and expecting it to work everywhere.

$$b(z_T) + z_T = \int dx_D \underset{z_T}{\text{argmax}}[p(x_D|z_T)]p(x_D|z_T)$$
$$= \int dx_D x_D \frac{1}{\sqrt{2\pi\epsilon^2}} e^{-\frac{(x_D - z_T)^2}{2\epsilon^2}}$$
$$= z_T. \quad (13)$$

Here, we have made use of the fact that for a Gaussian, $p(x_D|z_T)$ is maximized at $x_D = z_T$, and that the average of this Gaussian is simply z_T. This conclusion holds even if the detector response includes offsets, or if the noise ϵ depends on z_T.[5]

The strategy in Ref. [43] is to estimate the (local) likelihood density by extremizing the Donsker-Varadhan representation (DVR) [49,50] of the Kullback-Leibler divergence [51]:

$$L[f] = \mathbb{E}_{p(x_D,z_T)}[f(x_D, z_T)] - \log \mathbb{E}_{p(x_D)p(z_T)}[e^{f(x_D,z_T)}]. \quad (14)$$

By parametrizing $f(x_D, z_T)$ via a specially chosen Gaussian Ansatz (see Ref. [43] for details), one can extract the local maximum likelihood estimate and resolution with a single neural network training.

We focused on regression in the above discussion, but prior dependence also appears in classification calibration. A classifier trained with the MSE loss function or the binary cross entropy (BCE) will learn the probability of the signal given an observed x_D. If the fraction of the signal is different in the training set and the test set, that is, $p_{\text{test}}(z_T) \neq p_{\text{train}}(z_T)$, then the output can no longer be interpreted as the probability of the signal. Luckily, classifiers are almost never used this way in HEP, since the classification score is not interpreted directly as a probability.[6] In this case, simulation-based calibrations may not be required,[7] though data-based calibrations are still essential, as described next.

D. Data-based calibration

In data-based calibration, the goal is to account for possible differences between a true detector response, $p_{\text{data}}(x_D)$, and a simulated detector model, $p_{\text{sim}}(x_D)$. That is, the goal is to match detector level features x_D between data and a simulation at the distribution level, in contrast to simulation-based distribution, where the goal is to match x_D and a target feature z_T at the object level. Usually, $p_{\text{data}}(x_D)$ is a control dataset, and $p_{\text{sim}}(x_D) = \int dz_T p_{\text{sim}}(x_D|z_T)p_{\text{train}}(z_T)$ is a simulated detector output generated from truth-level features z_T.

In the machine learning literature, data-based calibration is called *domain adaptation*. Machine learning domain adaptation has been widely studied in the context of HEP [53–57] (see also *decorrelation* [58–74]), but these tools have not yet been applied to per-object calibrations. Traditional methods typically use binned or simple parametric approaches to calibrate differences between data and simulation.

The authors of Ref. [42] propose to use tools from the field of *optimal transport* (OT) to perform the data-based calibration using machine learning. The central idea is to learn a map $h: \mathbb{R}^N \to \mathbb{R}^N$ that "moves" x_D as little as possible, but still achieves $p_{\text{sim}}(x_D) \mapsto p_{\text{data}}(x_D)$. In this case, the OT-based calibration is

$$\hat{p}(x_D) = p_{\text{sim}}(h(x_D))|h'(x_D)|, \quad (15)$$

where $|h'(x_D)|$ is the Jacobian factor. The precise transportation map depends on the choice of OT metric. Equation (15) can be interpreted as shifting simulated samples x_D to $h(x_D)$, and additionally reweighting each sample by $|h'(x_D)|$. One can also write a corresponding expression for the OT-calibrated detector model, conditioned on z_T:

$$\hat{p}(x_D|z_T) = p_{\text{sim}}(h(x_D)|z_T)|h'(x_D)|. \quad (16)$$

Equation (16) can be thought of as a "corrected simulated response" function that accounts for mismodeling in the original simulation, $p_{\text{sim}}(x_D|z_T)$. At first glance, Eq. (16) might seem prior independent, since it is conditioned on the truth-level z_T. As we will see, though, there is implicit prior dependence in h. For simplicity, consider the special case of one dimension. Here, for any OT metric, the OT map $h: \mathbb{R} \to \mathbb{R}$ is simply given by

$$h(x_D) = P^{-1}_{\text{data}}(P_{\text{sim}}(x_D)), \quad (17)$$

where P_λ is the cumulative distribution function of λ, i.e., $P_\lambda(x_D) = \int_{-\infty}^{x_D} dx'_D p_\lambda(x'_D)$. This function maps quantiles of the simulated distribution to quantiles of the data distribution. The Jacobian of this transformation is

$$|h'(x_D)| = \frac{p_{\text{sim}}(x_D)}{p_{\text{data}}(h(x_D))}$$
$$= \frac{\int dz_T p_{\text{sim}}(x_D|z_T)p_{\text{train}}(z_T)}{p_{\text{data}}(h(x_D))}. \quad (18)$$

Thus, since the prior $p_{\text{train}}(z_T)$ explicitly appears, the derived OT-based detector model in Eq. (16) is prior dependent.

[5]It is not always true that a maximum likelihood calibration is unbiased. For instance, if X_D is drawn from a uniform distribution $U(0, z_T)$, then the maximum likelihood estimate from a single x_D sample is $\hat{z}_T = x_D$, whereas an unbiased estimate would be $\hat{z}_T = 2x_D$.

[6]See Ref. [52] for a review in the machine learning literature and Ref. [53] for related studies in the context of HEP likelihood ratios.

[7]There may be practical issues associated with prior dependence; e.g., if there is an extreme class imbalance, the classifier may not learn well. In the extreme limit of only one class present in the training, then there is a prior dependence also on the result.

In line with simulation-based calibration, the bias of a data-based calibration is the average difference between the estimator $\hat{p}(x_D)$ and the desired value $p_{\text{data}}(x_D)$, conditioned on x_T.[8] For OT-based calibration, the bias for a given value of x_D is

$$\begin{aligned} b(x_D) &= p_{\text{sim}}(h(x_D))|h'(x_D)| - p_{\text{data}}(x_D) \\ &= \int dz_T p_{\text{sim}}(h(x_D)|z_T) p_{\text{test}}(z_T)|h'(x_D)| \\ &\quad - p_{\text{data}}(x_D). \end{aligned} \tag{19}$$

If $p_{\text{test}}(z_T) = p_{\text{train}}(z_T)$, then the bias is zero. Otherwise, the calibration is biased, a consequence of prior dependence. Note that this is in contrast to simulation-based calibration, where nonuniversality can imply a bias even if $p_{\text{test}}(z_T) = p_{\text{train}}(z_T)$.

E. Unbiased data-based approaches?

As defined above, the goal of a data-based calibration is to match $p_{\text{sim}}(x_D)$ to $p_{\text{data}}(x_D)$. This is an inherently prior dependent task, however, since $\hat{p}(x_D) = \int dz_T \times \hat{p}(x_D|z_T) p_{\text{train}}(z_T)$—that is to say, the simulated detector output depends on the simulation input. Instead, one can ask if the corrected response function, $\hat{p}(x_D|z_T)$, is universal. If it is, then one can use the same corrected response function to generate $\hat{p}(x_D)$ for a variety of priors $p_{\text{test}}(z_T)$. At least in the special case of one-dimensional OT-based calibration, however, we have shown above that the corrected response function is *not* universal.

To our knowledge, no one has proposed a data-based calibration method that is prior independent, whether using machine learning or not. This implies that all data-based calibration methods in use are biased, though the degree of bias may be small if the testing and training truth-level densities are similar enough. We encourage the community to develop a prior-independent data-based calibration strategy, or prove that it is impossible.

III. RESOLUTION AND UNCERTAINTY IN CALIBRATIONS

The discussion thus far has focused on mitigating bias in calibration. Two related concepts are the resolution and uncertainty of a calibration. In this section, we review calibration resolution and uncertainty, and we clarify important nomenclature in HEP settings.

[8]This differs from the simulation-based calibration definition, which was conditioned on z_T. In data, there is no truth level z_T. However, sometimes, a proxy can be used as a z_T in data, allowing for a direct comparison of true versus reconstructed z_T values in data-based calibration. For example, when performing data-based calibration on a Z + jets sample, the p_T of the Z can be used as a proxy for the true jet p_T.

A. Resolution

As already mentioned, the bias of a calibration refers to the difference in central tendency (such as the mean, median, or mode) between a reconstructed quantity and a reference quantity. By contrast, the *resolution* of a calibration refers to the *spread* in the difference between the reconstructed and reference quantities. Using variance as our measure of spread, the resolution $\Sigma^2(z_T)$ can be written as the variance of differences between the reconstructed and truth values, conditioned on the truth values, evaluated over the test sample:

$$\Sigma^2(z_T) = \text{Var}_{\text{test}}[f(X_D) - z_T | Z_T = z_T]. \tag{20}$$

Resolution, like biases, can be prior dependent. When using the MSE-based calibration [Eq. (3)], this becomes

$$\begin{aligned} &\Sigma^2(z_T) + b^z(z_T) \\ &\quad = \int dx_D \left(\int dz_T z_T' p_{\text{train}}(z_T'|x_D) - z_T \right)^2 p_{\text{test}}(x_D|z_T). \end{aligned} \tag{21}$$

The prior dependence is seen by applying Bayes' theorem to $p_{\text{train}}(z_T'|x_D)$.

As before, this prior dependence can be reduced if the prior is wide compared to the detector response. If the prior $p(z_T)$ is Gaussian with standard deviation σ, and the detector response $p(x_D|z_T)$ is a Gaussian noise model with standard deviation ϵ, then by applying Eq. (22), one can show that the resolution scales as

$$\Sigma^2(z_T) \sim \epsilon^2 + \mathcal{O}\left(\left(\frac{\epsilon}{\sigma} \right)^4 \right) \epsilon^2. \tag{22}$$

On the other hand, for the prior-independent MLC calibration [Eq. (12)], the resolution can be shown to be

$$\Sigma^2(z_T) = \epsilon^2. \tag{23}$$

In HEP (and many other) applications, however, it is common to instead refer to the resolution with respect to a measurement x_D rather than the true value z_T. That is, for an inference $\hat{z}_T = f(x_D)$, we would like a measure of the spread of z_T values consistent with this measurement, which we will denote $\Sigma(x_D)$ (distinguished by the x_D argument rather than z_T). Depending on the context and type of calibration, there are a variety of ways to define $\Sigma(x_D)$—for instance, as the standard deviation from a Gaussian fit to the distribution of reconstructed over true energies (see, e.g., Ref. [45]). For our purposes, we can define the point resolution $\Sigma^2(x_D)$ as the variance of z_T's conditioned on x_D:

$$\Sigma^2(x_D) = \mathrm{Var}_{\mathrm{test}}[Z_T|X_D = x_D] = \mathbb{E}_{\mathrm{test}}[(f_{\mathrm{MSE}}(x_D) - Z_T)^2|X_D = x_D]. \quad (24)$$

For the MSE-based calibration, this is simply the variance of the posterior, $p(z_T|x_D)$. However, for frequentist approaches where the posterior is not well defined, such as the maximum likelihood calibration, the resolution cannot be defined this way and care must be taken. For Gaussian noise models $p(x_D|z_T)$, the likelihood is symmetric under interchanging the arguments x_D and z_T, so one can take the resolution to be [applying Eq. (20)]

$$\Sigma^2(x_D) = \Sigma^2(z_T) = \epsilon^2. \quad (25)$$

Calibrations do not necessarily improve the resolution and can sometimes make the resolution seem worse. For example, if a calibration requires multiplying the reconstructed quantity by a fixed number greater than one, then the resolution will grow by the same amount.[9] It is therefore important to compare resolutions only after calibration.

If a calibration incorporates many features that determine the resolution of a given quantity, then the resolution can improve from calibration. For example, suppose the reconstructed value x_D is some function of observable quantities $\vec{y}_D = (y_{D1}, y_{D2}, \ldots, y_{Dn})$, i.e., $x_D = g(\vec{y}_D)$. For instance, in the context of jet energy calibrations, $x_D = \alpha\eta$ for some constant α and an observable quantity η (e.g., energy dependence on the pseudorapidity). If any of the $\vec{y}_D$ have a nontrivial probability density, this will be inherited by the reconstructed value x_D, and thus x_D will have a nonzero resolution. This resolution is completely reducible, however, through a calibration that is $\vec{y}_D$ dependent—that is, a calibration function $\hat{z}_T = f'(\vec{y}_D)$ rather than $\hat{z}_T = f(x_D)$. The ability to incorporate many auxiliary features is why machine-learning-based approaches, such as the Gaussian Ansatz [43], have the potential to improve analyses at HEP experiments.

B. Uncertainty

In the machine learning literature, "resolution" would be referred to as a type of "uncertainty." Uncertainty in the statistical context refers to the limited information about z_T contained in x_D. In the HEP literature, though, we use uncertainty in a different way, to instead refer to the limited information we have about the bias and resolution of a calibration.

The reason for this difference in nomenclature is that HEP research is based primarily on simulation-based inference, where data are analyzed by comparison to model predictions. (This is the case for the vast majority of analyses at the LHC.) In this context, the word "uncertainty" is reserved to refer to uncertainties on model parameters. A worse resolution can degrade the statistical precision of a measurement, but if it is well modeled by the simulation, then there is no associated systematic uncertainty (though there will still be statistical uncertainties).

Both simulation-based and data-based calibrations can have associated uncertainties. For simulation-based calibrations, even if they are prior independent, there can be uncertainties in the detector models themselves. For data-based calibrations, there are additional uncertainties associated with the truth-level prior; see Sec. II E.

One of the goals of data-based calibration is to improve the modeling of the calibration in simulation to match the data. Typically, data-based calibrations are performed in dedicated event samples with well-understood physics processes. The residual uncertainty following the data-based calibration is dominated by the modeling of the underlying process. For example, data-based jet calibrations (called *in situ* calibrations) compare the jet to a well-measured reference object such as a Z boson. The momentum imbalance between the jet and the Z boson will be due in part to differences in the calibration between data and simulation and in part due to the mismodeling of initial and final state radiation. Uncertainties on the latter are then incorporated into the data-based calibration uncertainty. In nearly all cases, data-based calibrations are performed independently of the uncertainties, which are computed *post hoc*. In the future, these uncertainties may be improved with uncertainty/inference-aware machine learning methods [32,58–88].

IV. GAUSSIAN EXAMPLES

In this section, we demonstrate some of the calibration issues related to bias and prior dependence in a simple Gaussian example. We assume that the truth information (the "prior") is distributed according to a Gaussian distribution with mean μ and variance σ^2:

$$Z_T \sim \mathcal{N}(\mu, \sigma^2). \quad (26)$$

The detector response is assumed to induce Gaussian smearing centered on the truth input with variance ϵ^2:

$$X_D|Z_T = z_T \sim \mathcal{N}(z_T, \epsilon^2). \quad (27)$$

For the simulation-based calibration in Sec. IV A, the goal is to learn Z_T given X_D, assuming perfect knowledge of the detector response. For the data-based calibration in Sec. IV B, the goal is to map X_D in "simulation" to X_D in "data." In this latter study, we assume that data and simulation have the same true probability density and differ only in their detector response, $\epsilon_{\mathrm{sim}} \neq \epsilon_{\mathrm{data}}$—that is, p_{sim} "mismodels" p_{data}.

[9] This is also true if we had used the relative resolution, $\mathbb{E}[\frac{f(x_D)}{z_T}|Z_T = z_T]$, which is also commonly used in HEP, rather than the absolute resolution.

A. Simulation-based calibration

If we use the MSE approach in Eq. (3), there is a prior dependence in the calibration, which induces bias. Perhaps counterintuitively, this bias persists even if the prior is the same as the data density,

$$p_{\text{train}} = p_{\text{test}} \equiv p, \tag{28}$$

as we now show.

In the Gaussian case, the reconstructed data are distributed according to

$$X_D \sim \mathcal{N}(\mu, \sigma^2 + \epsilon^2), \tag{29}$$

and it is possible to solve Eq. (5) analytically, in the asymptotic limit

$$f_{\text{MSE}}(x_D) = \frac{\epsilon^2\mu + \sigma^2 x_D}{\epsilon^2 + \sigma^2}. \tag{30}$$

For comparison, we can also compute the unbiased maximum likelihood calibration using Eq. (12):

$$f_{\text{MLC}}(x_D) = x_D. \tag{31}$$

It is also possible to analytically compute the point resolutions, $\Sigma(x_D)$, for both the MSE and the MLC fits [Eqs. (24) and (25), respectively]:

$$\Sigma_{\text{MSE}}(x_D) = \frac{\epsilon\sigma}{\sqrt{\epsilon^2 + \sigma^2}}, \tag{32}$$

$$\Sigma_{\text{MLC}}(x_D) = \epsilon. \tag{33}$$

To illustrate this setup, we simulate this scenario numerically for $\mu = 0$, $\sigma = 1$, and $\epsilon = 2$. In Fig. 1(a), we show the simulated data, for which both the true and reconstructed values follow a Gaussian distribution. The first step of a typical calibration is to predict the true z_T from the reconstructed x_D. Since we know that the average dependence of the true z_T on the reconstructed x_D is linear, we perform a first-order polynomial fit to the data using NumPy POLYFIT, which is represented by the blue dashed line in Fig. 1(a). This calibration function is then applied to all reconstructed values:

$$\hat{z}_T(x_D) = f_{\text{MSE}}(x_D). \tag{34}$$

The resulting calibration curve is presented in blue in Fig. 1(b), along with the associated resolution $\Sigma_{\text{MSE}}(x_D)$.

For comparison, we perform a maximum likelihood calibration using the Gaussian Ansatz introduced in Ref. [43]:

$$f(x,z) = A(x) + (z - B(x)) \cdot D(x) + \frac{1}{2}(z - B(x))^T \cdot C(x,z) \cdot (z - B(x)), \tag{35}$$

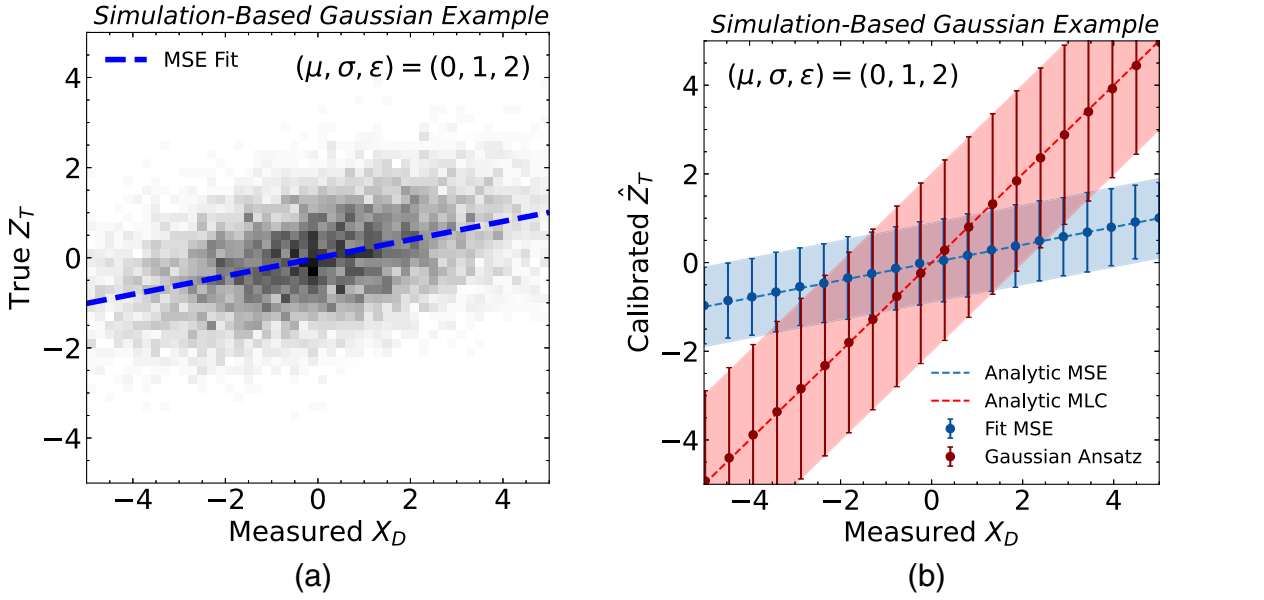

FIG. 1. (a) Two-dimensional histogram of the reconstructed value x_D distribution versus the true value z_T distribution, in the Gaussian example with $\mu = 0$, $\sigma = 1$, and $\epsilon = 2$. The dashed line represents a linear fit to the data points. (b) For test values of x_D, the vertical axis is the calibrated target value $\hat{z}_T(x_D)$. The blue dots are the results from a numerical MSE fit $f_{\text{MSE}}(x_D)$, and the error bars correspond to the numerical point resolution $\Sigma_{\text{MSE}}(x_D)$, with the analytic prediction in the red dotted line. For comparison, the Gaussian Ansatz calibration is indicated by the red points $f_{\text{MLC}}(x_D)$, with the error bars indicating the point resolution $\Sigma_{\text{MLC}}(x_D)$. For both fits, the colored lines and bands are the analytically expected results for the fits and resolutions, respectively.

where we have dropped the subscripts ($x_D \to x$, $z_T \to z$) for compactness of notation. As described in Ref. [43], the calibration function $B(x)$ is obtained by minimizing the DVR loss function from Eq. (14), such that after training

$$\hat{z}_T(x_D) = B(x_D), \tag{36}$$

$$\Sigma_{\mathrm{MLC}}(x_D) = -[C(x_D, B(x_D))]^{-1/2}. \tag{37}$$

For Gaussian noise models, this maximum likelihood estimate is unbiased, as confirmed by the numerical results in Fig. 1(b). We implement the Gaussian Ansatz in KERAS [89] with the TensorFlow backend [90]. The A network consists of three hidden layers with 16 nodes per layer, with rectified linear unit activations. The B and C networks are each a single node with linear activation. The D network is set to zero by hand. Optimization is carried out with ADAM [91] over 100 epochs with a batch size of 128. As desired, the Gaussian Ansatz yields a calibration that is independent of the prior $p_{\mathrm{train}}(z_T)$.

To demonstrate the bias, we plug Eq. (28) into Eq. (8) to get the bias from the MSE calibration approach:

$$b(z_T) + z_T = \int dx_D dz_T' z_T' p(x_D|z_T') p(x_D|z_T) \frac{p(z_T')}{p(x_D)}. \tag{38}$$

It is possible to solve Eq. (38) analytically for the Gaussian setup:

$$b(z_T) = \left(\frac{\epsilon^2}{\sigma^2 + \epsilon^2}\right)(\mu - z_T). \tag{39}$$

As expected, $b(z_T) \to 0$ as $\epsilon \to 0$. For $\epsilon > 0$, though, there is a nonzero bias with the MSE approach. The z_T-binned resolutions can also be computed using Eqs. (22) and (23):

$$\Sigma_{\mathrm{MSE}}(z_D) = \frac{\sigma^2}{\epsilon^2 + \sigma^2}\epsilon, \tag{40}$$

$$\Sigma_{\mathrm{MLC}}(z_D) = \epsilon. \tag{41}$$

The fitted biases and resolutions are presented in Fig. 2, which exhibits the bias expected from Eq. (39). This illustrates the large bias introduced by the MSE regression procedure.

To further highlight the role of prior dependence, we repeat the MSE calibration procedure, where we test multiple values of the prior parameters μ and σ to confirm the predictions in Eq. (39). As shown in Fig. 2(a), changes in μ simply shift the calibration up and down, but do not improve the calibration quality across the true values of z_T. As shown in Fig. 2(b), changes in σ change the slope of the calibration. In the limit $\sigma \to \infty$, the calibration curve approaches the unbiased curve, as anticipated from Eq. (9).

B. Data-based calibration

As discussed in Sec II E, we are unaware of any prior-independent data-based calibration. To highlight this challenge, we study the OT-based technique introduced in Ref. [42] and mentioned in Sec. II. D. In our Gaussian example, the goal is to calibrate a "simulation" sample with $(\mu_{\mathrm{sim.}}, \sigma_{\mathrm{sim.}}, \epsilon_{\mathrm{sim.}})$ to match a "data" sample with $(\mu_{\mathrm{data}}, \sigma_{\mathrm{data}}, \epsilon_{\mathrm{data}})$.

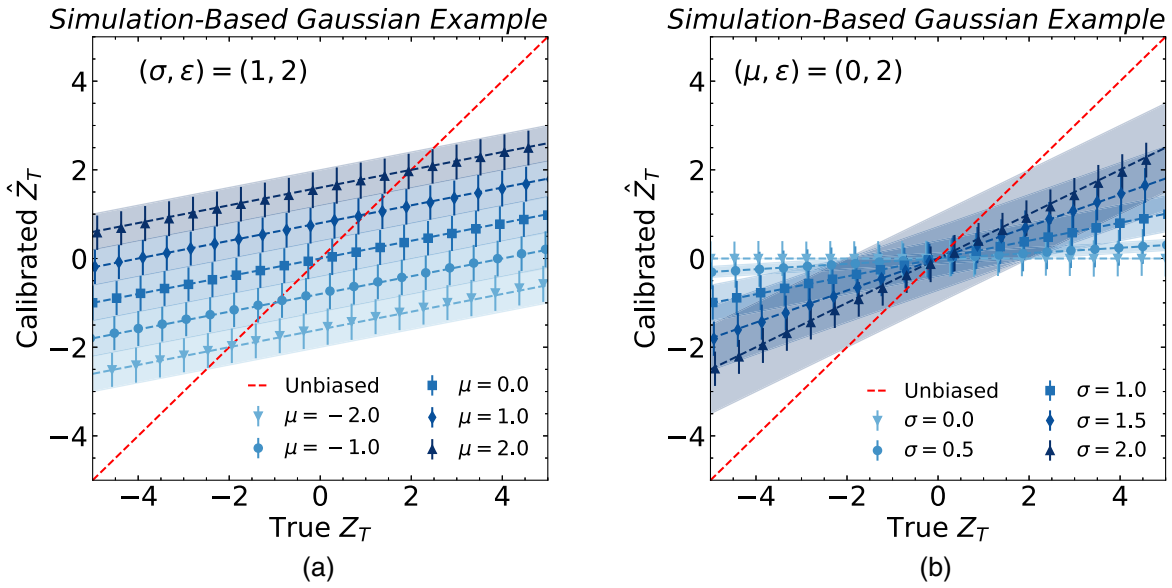

FIG. 2. The same MSE results as Fig. 1(b), but plotted in bins of true z_T rather than x_D. Points correspond to numerical fit results with associated resolution $\Sigma_{\mathrm{MSE}}(z_T)$, while the dashed lines and bands correspond to analytic results. Multiple values of the prior parameters (a) μ and (b) σ are shown to illustrate the prior dependence of the bias. Though not shown, we verified that the Gaussian Ansatz gives results consistent with the unbiased calibration in dashed red lines.

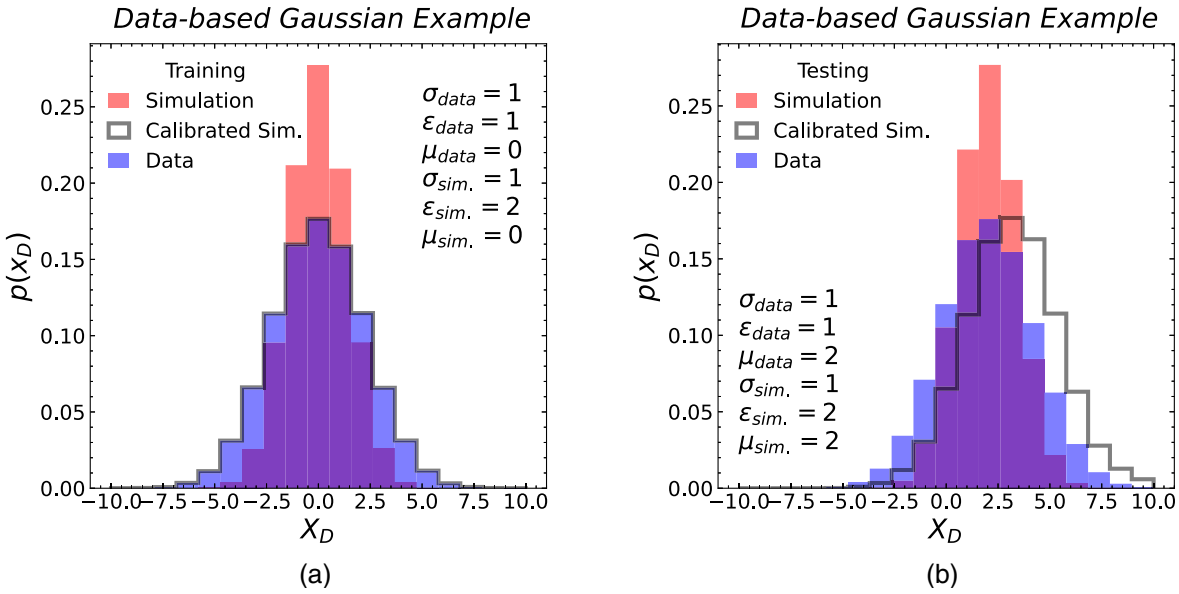

FIG. 3. Histograms of the raw simulation, calibrated simulation, and data for (a) the training set and (b) the test set, the Gaussian example of data-based calibration. The calibration function for the test set is used in both figures.

For simplicity, we assume that the true spectra [determined by (μ, σ)] are the same in data and in simulation, such that there is no systematic uncertainty in the calibration (see Sec. III B). Only ϵ, the parameter governing the detector response, is different between simulation and data—the simulation mismodels the real detector. To highlight the issue of prior dependence, we consider a "training" set with one value of $\mu_{\text{train}} = 0$ and a "testing" set with a different value of μ_{test}, with a shared value of σ. The calibration will be derived on the training set and deployed on the testing set. Again for simplicity, we assume that detector effects (determined by ϵ) are the same in both the train and the test sets.

The one-dimensional OT map h from one Gaussian A to another Gaussian B can be computed analytically:

$$h_{A\to B}(x) = \frac{x - \mu_A}{\sigma_A} \cdot \sigma_B + \mu_B, \tag{42}$$

where the mean and standard deviation of sample i are μ_i and σ_i, respectively. This equation can be derived following Eq. (17), by computing cumulative distribution function (CDF) of sample A with the inverse CDF of sample B.

For the training set with $\mu_{\text{train}} = 0$, we have

$$h_{\text{train}}(x) = \frac{\sqrt{\sigma^2 + \epsilon^2_{\text{data}}}}{\sqrt{\sigma^2 + \epsilon^2_{\text{sim}}}} x \equiv \alpha x. \tag{43}$$

The test set only differs in the value of μ_{test}, so the correct calibration function should be

$$h_{\text{test}}(x) = \alpha(x - \mu_{\text{test}}) + \mu_{\text{test}} = \alpha x + \mu_{\text{test}}(1 - \alpha). \tag{44}$$

As long as $\alpha \neq 1$, then $h_{\text{train}} \neq h_{\text{test}}$, and so the calibration is not universal.

A numerical demonstration of this bias is presented in Fig. 3, where histograms of the data and simulation are presented along with the calibrated result. In Fig. 3(a), we see the calibration derived in the training sample, where by construction, the calibrated simulation matches the data. Since the truth distribution is different in the test set, however, the training calibration applied in the test set is biased, as shown in Fig. 3(b). The actual calibration function is plotted in Fig. 4 and compared to the analytic expectation from Eqs. (44) and (43). The fact that the calibration derived on the

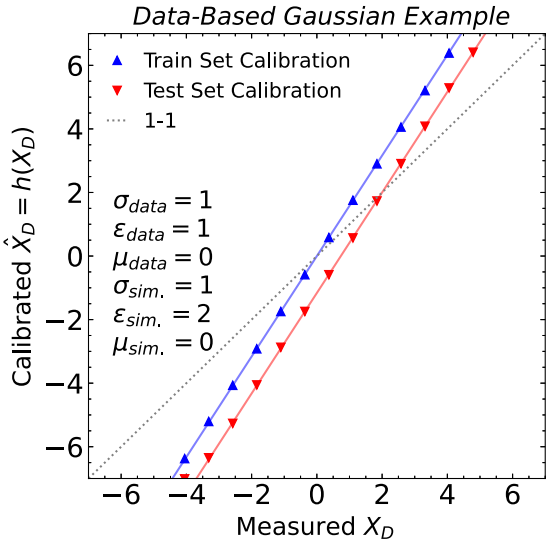

FIG. 4. The data-driven calibration functions corresponding to Fig. 3. The blue points correspond to the calibration function h_{train} derived from the training set, and the red points correspond to the ideal calibration h_{test} one would derive from the test set.

train set is not the same as the calibration derived on the test set shows that the calibration derived in one and applied to the other will lead to a residual bias.

V. CALIBRATING JET ENERGY RESPONSE

Jets are ubiquitous at the LHC, and their calibration is an essential input to a majority of physics analyses performed by ATLAS and CMS. In this section, we consider a simplified version of simulation-based and data-based jet energy calibrations. To illustrate the impact of the prior dependence, we use a realistic and also extreme example where calibrations are derived in a sample of generic quark and gluon jets and then applied to a test sample of jets from the decay of a heavy new resonance. To further simplify the problem, we consider a calibration of the invariant mass m_{jj} of the leading two jets. In practice, jet energy calibrations are derived for individual jets, but this requires at least including calibrating the jet rapidity in addition to the jet energy. We keep the problem one-dimensional in order to ensure the problem is easy to visualize and to mitigate the dependence on features that are not explicitly modeled. For a high-dimensional study of jet energy calibrations in a prior-independent way, see Ref [43].

A. Datasets

Our study is based on generic dijet production in quantum chromodynamics (QCD). For these studies we will consider two different datasets to demonstrate simulation-based and data-based jet energy calibrations. The first dataset is made with a full detector simulation. The full simulation sample uses PYTHIA6.426 [92] with the Z2 tune [93] and interfaced with a GEANT4-based [94–96] full simulation of the CMS experiment [97]. In simulation-based calibration, our goal will be to reconstruct the truth-level $z_T = m_{jj}^{\text{true}}$ from the detector-level $x_D = m_{jj}^{\text{reco}}$. The second dataset is constructed with a fast detector simulation. The fast simulation uses PYTHIA8.219 [98] interfaced with DELPHES3.4.1 [99–101] using the default CMS detector card. In data-based calibration, our goal will be to match this fast simulation to "data," which will be represented by the full simulation. The full simulation sample comes from the CMS Open Data Portal [102–104] and processed into an MIT Open Data format [105–108]. The fast simulation sample is available at Refs. [109,110].

For each dataset, we have access to the parton-level hard-scattering scale $\hat{p}_T$ from PYTHIA, which is in general different from the jet-level transverse momentum p_T we are interested in studying. To avoid any issues related to the trigger, we focus on events where $\hat{p}_T > 1$ TeV. Particles (at truth level) or particle flow candidates (at the reconstructed level) are used as inputs to jet clustering, implemented using FastJet 3.2.1 [111,112] and the anti-k_t algorithm [113] with radius parameter $R = 0.5$. No calibrations are applied to the reconstructed jets.

To emulate two different physics processes while controlling for all hidden variables, we consider dijet events with two different sets of event weights. This will allow us to study the prior-dependent effects of each calibration.

(i) *QCD*. This set of weights $\{w_i\}$ comes from the original PYTHIA event generation. The resulting spectra are steeply falling in the invariant mass of the two jets, m_{jj}.

(ii) *Beyond the Standard Model (BSM)*. To emulate a narrow dijet resonance, we consider a second set of weights given by

$$w(m_{jj,i}^{\text{true}}) \propto \frac{1}{\sigma w_i} \exp\left[-\frac{(m_{jj,i}^{\text{true}} - \mu)^2}{2\sigma^2}\right], \quad (45)$$

where $\mu = 2.8$ TeV and $\sigma = 10$ GeV. Note that the weighting is applied using the true m_{jj}.

The m_{jj} distributions as described above are shown in Fig. 5. In the full simulation, one can see a difference between m_{jj}^{true} and m_{jj}^{reco} for both QCD and BSM, necessitating a simulation-based calibration. Additionally, the m_{jj}^{reco} distribution is significantly different between the full and fast simulations, which to correct requires a data-based calibration.

For all following results, half of the examples are used for training and half are used for testing.

B. Simulation-based calibration

The goal for the simulation-based calibration is to learn a function to predict $z_T = m_{jj}^{\text{true}}$ from $x_D = m_{jj}^{\text{reco}}$ in the full simulation. In contrast to the Gaussian example in Sec. IV A,

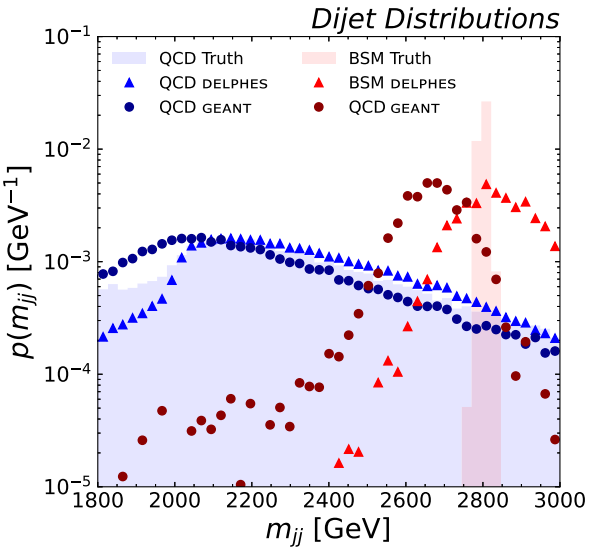

FIG. 5. The m_{jj} distributions for QCD (blue) and BSM (red) events in the fast and full simulation. The shaded histograms correspond to the $z_T = m_{jj}^{\text{true}}$ truth-level distributions, whereas the light triangles and dark circles correspond to $x_D = m_{jj}^{\text{reco}}$ for the fast (DELPHES) and slow (GEANT4) distributions, respectively.

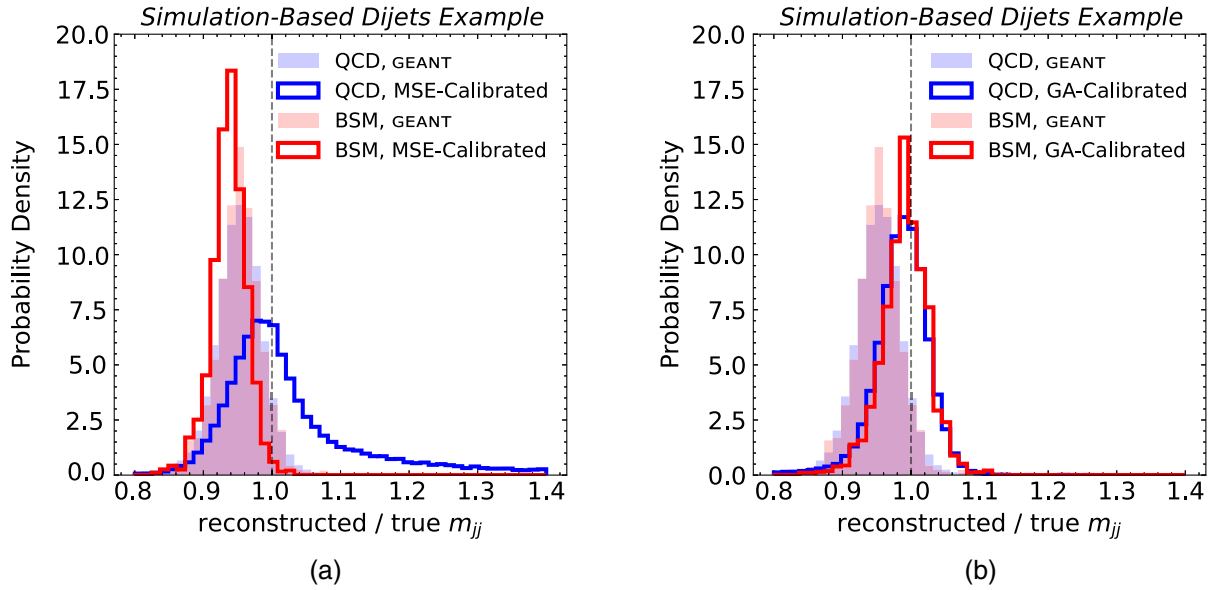

FIG. 6. The reconstructed m_{jj} divided by the true m_{jj} for the QCD and BSM samples, using (a) the MSE-based approach and (b) the maximum likelihood approach with the Gaussian Ansatz. Shown are results with and without the simulation-based calibration applied.

we do not know the functional form of the calibration. Therefore, we use a neural network to provide a flexible parametrization of the calibration and numerically minimize the MSE loss. The neural network has three hidden layers with 50 nodes per layer, with the rectified linear unit activation for intermediate layers and a linear activation for the output. The network is implemented in KERAS with the TensorFlow backend and optimized with ADAM using a batch size of 1000 and 50 epochs. Training is performed over the QCD sample to obtain the calibration function. The learned calibration function is then applied to both the QCD and BSM test samples.

The result of MSE calibration is shown in Fig. 6(a). Prior to any calibration, the detector response is about 5% low in both the QCD and BSM test samples. After calibration, the mean is nearly unity for the QCD sample, albeit with a large width—that is to say, the average bias is close to zero over the prior, but the average resolution is large. For the BSM sample, though, the calibrated mean is far from unity, demonstrating the bias and prior dependence of the MSE calibration. The MSE-based calibration obtained from the QCD fit is not universal and gives poor results when applied to the BSM sample.[10]

For comparison, in Fig. 6(b) we show results from a maximum-likelihood-based calibration trained on the QCD sample, using the Gaussian Ansatz in Eq. (35). The A, B, C, and D networks of the Gaussian Ansatz each consist of three hidden layers with 32 nodes per layer, with the same activation functions, batch size, and epochs as in the Gaussian example. The calibration function trained on the QCD sample can be used for the BSM sample, and as Fig. 6(b) shows, the calibration is indeed universal and unbiased, as expected.

C. Data-based calibration

The goal for the data-based calibration task is to "correct" $p_{\text{sim}}(m_{jj}^{\text{reco}})$, given by the fast simulation (DELPHES), to the observed data distribution $p_{\text{data}}(m_{jj}^{\text{reco}})$, given by the full simulation (GEANT4). We now apply the same procedure described in Sec. IV B to the dijet example.

An OT-based calibration is derived using QCD jets, to align the fast simulation DELPHES sample with the full simulation GEANT4 sample. The calibration function, given by the optimal transport map [Eq. (17)], can be computed numerically by sorting and integrating the weighted data points to build the cumulative distribution functions. On the QCD sample, this calibration closes by construction. In particular, as shown in Fig. 7(a), the blue dashed line in the ratio plot fluctuates around unity, with deviations due to statistical fluctuations that differ between the two halves of the event samples.

When this calibration is applied to the BSM events, however, the calibration overshoots, as shown with the red dashed line in the ratio plot in Fig. 7(b). While the resulting dashed distribution agrees better with the data histogram in dark red than does the fast sim histogram in light red, the overall agreement is still rather poor. This again highlights the issue of prior dependence in data-based calibrations.

[10]The converse is also true—attempting to use a calibration fitted on the BSM sample will lead to bias on the QCD sample, or any other BSM sample for that matter. These nonuniversal fits lead to *mass sculpting*, in which a fit depends strongly on the mass point used in training. See, e.g., [114] for discussions on sculpting and mass decorrelation.

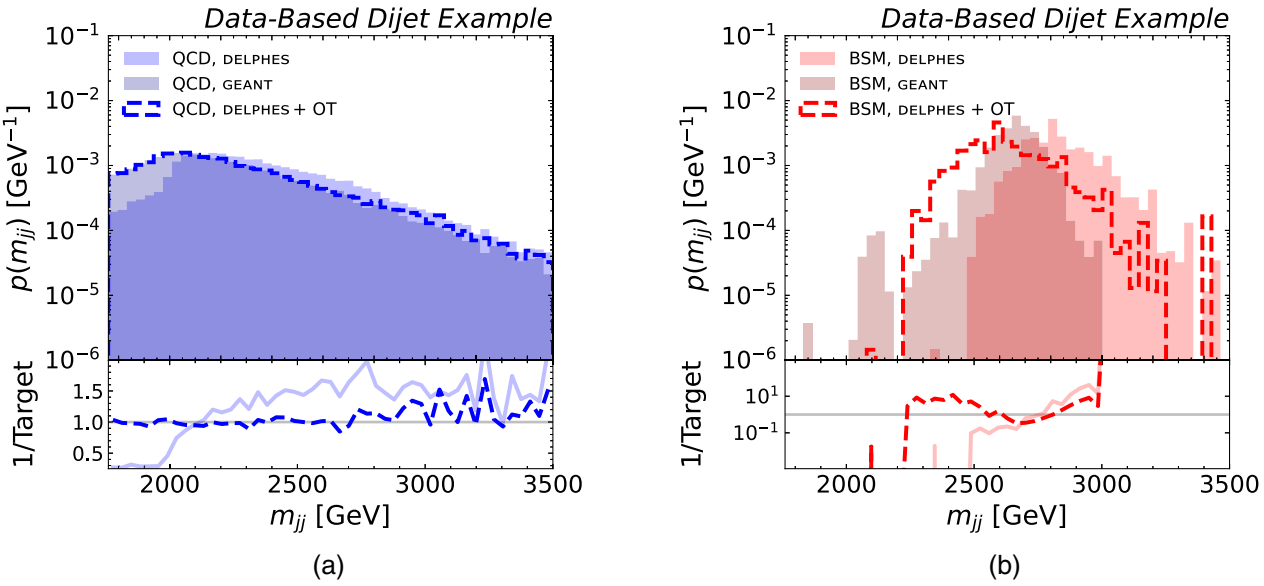

FIG. 7. The reconstructed m_{jj} for (a) QCD and (b) BSM events in the fast and full simulation, with and without the data-based OT calibration. The calibration is performed on the QCD sample, which closes, and the same calibration is applied to the BSM sample. Note that for the BSM sample, the ratio plot is in log-scale, indicating a very large bias.

VI. CONCLUSIONS

In this paper, we explored the prior dependence of machine-learning-based calibration techniques. There is a growing number of machine learning proposals for simulation-based and data-based calibration, and in nearly all cases, there is a prior dependence. We highlighted the resulting calibration bias in a synthetic Gaussian example and a more realistic particle physics example of dijet production at the LHC.

In the simulation-based calibration case, most proposals learn a truth target from detector-level observables using loss functions such as the MSE. A neural network trained in this way will learn the average true value given the detector-level inputs, which depends on the spectrum of truth values. However, we have shown that this will yield a calibration that lacks the critical properties of universality and closure.

There are fewer proposals for machine learning data-based calibrations, but we studied one recent idea based on OT and showed its prior dependence. While we focused on one-dimensional examples, the prior dependence is a generic feature of these approaches. Going to higher dimensions may even exacerbate the issue since it is harder to visualize and control prior differences in many dimensions.

New learning approaches are required to ensure that machine learning-based calibrations are universal. For simulation-based calibration, the ATLAS Collaboration has proposed a prior-independent method called *generalized numerical inversion* [23,24]. While prior independent, this technique is typically biased and does not scale well to many dimensions. We proposed a new approach based on maximum likelihood estimation in Ref [43], based on parametrizing the log-likelihood with a Gaussian Ansatz. Maximum-likelihood-based approaches are prior independent by construction and are well-motivated statistically. Parametrizing the maximum likelihood estimator with neural networks requires a different learning paradigm than current approaches, but it extends well to many dimensions. To our knowledge, there are currently no prior-independent data-based calibration approaches.

To make the most use of the complex data from the LHC and other HEP experiments, it is essential to use all of the available information for object calibration. This will require modern machine learning to account for all of the subtle correlations in high dimensions. It is important, however, that we construct these machine learning calibration functions in a way that integrates all of the features of classical calibration methods. We highlighted prior independence in this paper as a cornerstone of calibration. In the future, innovations that incorporate knowledge of the detector response or physics symmetries may further enhance the precision and accuracy of machine learning calibrations.

The code for this paper can be found at [115], which makes use of JUPYTER notebooks [116] employing NumPy [117] for data manipulation and MATPLOTLIB [118] to produce figures. All of the machine learning was performed on a Nvidia RTX6000 Graphical Processing Unit (GPU). The physics datasets are hosted on ZENODO at Refs. [106–108,110].

ACKNOWLEDGMENTS

B. N. is supported by the U.S. Department of Energy (DOE), Office of Science under Contract No. DE-AC02-05CH11231. R. G. and J. T. are supported by the National Science Foundation under Cooperative Agreement PHY-2019786 (The NSF AI Institute for Artificial Intelligence and Fundamental Interactions, [119]), and by the U.S. DOE Office of High Energy Physics under Grant No. DE-SC0012567.

[1] Georges Aad *et al.* (ATLAS Collaboration), Topological cell clustering in the ATLAS calorimeters and its performance in LHC Run 1, Eur. Phys. J. C **77**, 490 (2017).
[2] A. M. Sirunyan *et al.* (CMS Collaboration), Particle-flow reconstruction and global event description with the CMS detector, J. Instrum. **12**, P10003 (2017).
[3] Georges Aad *et al.* (ATLAS Collaboration), Jet energy scale and resolution measured in proton–proton collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector, Eur. Phys. J. C **81**, 689 (2021).
[4] Vardan Khachatryan *et al.* (CMS Collaboration), Jet energy scale and resolution in the CMS experiment in pp collisions at 8 TeV, J. Instrum. **12**, P02014 (2017).
[5] Georges Aad *et al.* (ATLAS Collaboration), Muon reconstruction performance of the ATLAS detector in proton–proton collision data at $\sqrt{s} = 13$ TeV, Eur. Phys. J. C **76**, 292 (2016).
[6] Albert M. Sirunyan *et al.* (CMS Collaboration), Performance of the reconstruction and identification of high-momentum muons in proton-proton collisions at $\sqrt{s} = 13$ TeV, J. Instrum. **15**, P02027 (2020).
[7] Georges Aad *et al.* (ATLAS Collaboration), Electron and photon performance measurements with the ATLAS detector using the 2015–2017 LHC proton-proton collision data, J. Instrum. **14**, P12006 (2019).
[8] Vardan Khachatryan *et al.* (CMS Collaboration), Performance of photon reconstruction and identification with the CMS detector in proton-proton collisions at sqrt$(s) = 8$ TeV, J. Instrum. **10**, P08010 (2015).
[9] Vardan Khachatryan *et al.* (CMS Collaboration), Performance of electron reconstruction and selection with the CMS detector in proton-proton collisions at $\sqrt{s} = 8$ TeV, J. Instrum. **10**, P06005 (2015).
[10] Georges Aad *et al.* (ATLAS Collaboration), Identification and energy calibration of hadronically decaying tau leptons with the ATLAS experiment in pp collisions at $\sqrt{s} = 8$ TeV, Eur. Phys. J. C **75**, 303 (2015).
[11] A. M. Sirunyan *et al.* (CMS Collaboration), Performance of reconstruction and identification of τ leptons decaying to hadrons and ν_τ in pp collisions at $\sqrt{s} = 13$ TeV, J. Instrum. **13**, P10005 (2018).
[12] Georges Aad *et al.* (ATLAS Collaboration), ATLAS b-jet identification performance and efficiency measurement with $t\bar{t}$ events in pp collisions at $\sqrt{s} = 13$ TeV, Eur. Phys. J. C **79**, 970 (2019).
[13] A. M. Sirunyan *et al.* (CMS Collaboration), Identification of heavy-flavour jets with the CMS detector in pp collisions at 13 TeV, J. Instrum. **13**, P05011 (2018).
[14] Morad Aaboud *et al.* (ATLAS Collaboration), Performance of top-quark and W-boson tagging with ATLAS in Run 2 of the LHC, Eur. Phys. J. C **79**, 375 (2019).
[15] Albert M. Sirunyan *et al.* (CMS Collaboration), Identification of heavy, energetic, hadronically decaying particles using machine-learning techniques, J. Instrum. **15**, P06005 (2020).
[16] Dawit Belayneh *et al.*, Calorimetry with deep learning: Particle simulation and reconstruction for collider physics, Eur. Phys. J. C **80**, 688 (2020).
[17] ATLAS Collaboration, Deep learning for pion identification and energy calibration with the ATLAS detector, Report No. ATL-PHYS-PUB-2020-018, 2020.
[18] N. Akchurin, C. Cowden, J. Damgov, A. Hussain, and S. Kunori, On the use of neural networks for energy reconstruction in high-granularity calorimeters, J. Instrum. **16**, P12036 (2021).
[19] N. Akchurin, C. Cowden, J. Damgov, A. Hussain, and S. Kunori, Perspectives on the calibration of CNN energy reconstruction in highly granular calorimeters, arXiv: 2108.10963.
[20] L. Polson, L. Kurchaninov, and M. Lefebvre, Energy reconstruction in a liquid argon calorimeter cell using convolutional neural networks, J. Instrum. **17**, P01002 (2022).
[21] Joosep Pata, Javier Duarte, Jean-Roch Vlimant, Maurizio Pierini, and Maria Spiropulu, MLPF: Efficient machine-learned particle-flow reconstruction using graph neural networks, Eur. Phys. J. C **81**, 381 (2021).
[22] Jan Kieseler, Giles C. Strong, Filippo Chiandotto, Tommaso Dorigo, and Lukas Layer, Calorimetric measurement of multi-TeV muons via deep regression, Eur. Phys. J. C **82**, 79 (2022).
[23] ATLAS Collaboration, Generalized numerical inversion: A neural network approach to jet calibration, Report No. ATL-PHYS-PUB-2018-013, 2018.
[24] ATLAS Collaboration, Simultaneous jet energy and mass calibrations with neural networks, Report No. ATL-PHYS-PUB-2020-001, 2020.
[25] Albert M. Sirunyan *et al.* (CMS Collaboration), A deep neural network for simultaneous estimation of b jet energy and resolution, Comput. Softw. Big Sci. **4**, 10 (2020).

[26] Rüdiger Haake and Constantin Loizides, Machine learning based jet momentum reconstruction in heavy-ion collisions, Phys. Rev. C **99**, 064904 (2019).
[27] Rüdiger Haake (ALICE Collaboration), Machine learning based jet momentum reconstruction in Pb-Pb collisions measured with the ALICE detector, Proc. Sci. EPS-HEP2019 (**2020**) 312 [arXiv:1909.01639].
[28] Pierre Baldi, Lukas Blecher, Anja Butter, Julian Collado, Jessica N. Howard, Fabian Keilbach, Tilman Plehn, Gregor Kasieczka, and Daniel Whiteson, How to GAN higher jet resolution, arXiv:2012.11944.
[29] Patrick T. Komiske, Eric M. Metodiev, Benjamin Nachman, and Matthew D. Schwartz, Pileup Mitigation with Machine Learning (PUMML), J. High Energy Phys. 12 (2017) 051.
[30] ATLAS Collaboration, Convolutional neural networks with event images for pileup mitigation with the ATLAS detector, Technical Report No. ATL-PHYS-PUB-2019-028, CERN, Geneva, 2019.
[31] B Maier, S M Narayanan, G de Castro, M Goncharov, Ch Paus, and M Schott, Pile-up mitigation using attention, Mach. Learn. **3**, 025012 (2022).
[32] Gregor Kasieczka, Michel Luchmann, Florian Otterpohl, and Tilman Plehn, Per-object systematics using deep-learned calibration, SciPost Phys. **9**, 089 (2020).
[33] J. Arjona Martínez, Olmo Cerri, Maurizio Pierini, Maria Spiropulu, and Jean-Roch Vlimant, Pileup mitigation at the Large Hadron Collider with graph neural networks, Eur. Phys. J. Plus **134**, 333 (2019).
[34] Markus Diefenthaler, Abduhhal Farhat, Andrii Verbytskyi, and Yuesheng Xu, Deeply learning deep inelastic scattering kinematics, arXiv:2108.11638.
[35] Junze Liu, Jordan Ott, Julian Collado, Benjamin Jargowsky, Wenjie Wu, Jianming Bian, and Pierre Baldi (DUNE Collaboration), Deep-learning-based kinematic reconstruction for DUNE, arXiv:2012.06181.
[36] S. Delaquis *et al.* (EXO Collaboration), Deep neural networks for energy and position reconstruction in EXO-200, J. Instrum. **13**, P08023 (2018).
[37] Pierre Baldi, Jianming Bian, Lars Hertel, and Lingge Li, Improved energy reconstruction in NOvA with regression convolutional neural networks, Phys. Rev. D **99**, 012011 (2019).
[38] R. Abbasi *et al.*, A convolutional neural network based cascade reconstruction for the IceCube Neutrino Observatory, J. Instrum. **16**, P07041 (2021).
[39] M. G. Aartsen *et al.* (IceCube Collaboration), Cosmic ray spectrum from 250 TeV to 10 PeV using IceTop, Phys. Rev. D **102**, 122001 (2020).
[40] Kiara Carloni, Nicholas W. Kamp, Austin Schneider, and Janet M. Conrad, Convolutional neural networks for shower energy prediction in liquid argon time projection chambers, J. Instrum. **17**, P02022 (2022).
[41] Matthew Feickert and Benjamin Nachman, A living review of machine learning for particle physics, arXiv:2102.02770.
[42] Chris Pollard and Philipp Windischhofer, Transport away your problems: Calibrating stochastic simulations with optimal transport, Nucl. Instrum. Methods Phys. Res., Sect. A **1027**, 166119 (2022).
[43] Rikab Gambhir, Benjamin Nachman, and Jesse Thaler, companion Letter, Learning Uncertainties the Frequentist Way: Calibration and Correlation in High Energy Physics, Phys. Rev. Lett. **129**, 082001 (2022).
[44] Sanha Cheong, Aviv Cukierman, Benjamin Nachman, Murtaza Safdari, and Ariel Schwartzman, Parametrizing the detector response with neural networks, J. Instrum. **15**, P01030 (2020).
[45] A. Cukierman and B. Nachman, Mathematical properties of numerical inversion for jet calibrations, Nucl. Instrum. Methods Phys. Res., Sect. A **858**, 1 (2017).
[46] Danilo Jimenez Rezende and Shakir Mohamed, Variational inference with normalizing flows, Int. Conf. Mach. Learn. **37**, 1530 (2015).
[47] Ivan Kobyzev, Simon Prince, and Marcus Brubaker, Normalizing flows: An introduction and review of current methods, IEEE Trans. Pattern Anal. Mach. Intell. **43**, 3964 (2020).
[48] Benjamin Nachman and Jesse Thaler, E pluribus unum ex machina: Learning from many collider events at once, Phys. Rev. D **103**, 116013 (2021).
[49] Monroe D. Donsker and S. R. S. Varadhan, Asymptotic evaluation of certain markov process expectations for large time, Commun. Pure Appl. Math. **28**, 1 (1975).
[50] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R. Devon Hjelm, Mine: Mutual information neural estimation, arXiv:1801.04062.
[51] Solomon Kullback and Richard A Leibler, On information and sufficiency, Ann. Math. Stat. **22**, 79 (1951).
[52] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger, On calibration of modern neural networks, in *Proceedings of the 34th International Conference on Machine Learning*, Proceedings of the Machine Learning Research, edited by Doina Precup and Yee Whye Teh (PMLR, 2017), Vol. 70, pp. 1321–1330.
[53] Kyle Cranmer, Juan Pavez, and Gilles Louppe, Approximating likelihood ratios with calibrated discriminative classifiers, arXiv:1506.02169.
[54] A. Rogozhnikov, Reweighting with boosted decision trees, in *Proceedings of the 17th International Workshop on Advanced Computing and Analysis Techniques in Physics Research (ACAT 2016): Valparaiso, Chile, 2016* (IOP, Pennsylvania, 2016), Vol. 762, p. 012036.
[55] Anders Andreassen and Benjamin Nachman, Neural networks for full phase-space reweighting and parameter tuning, Phys. Rev. D **101**, 091901 (2020).
[56] S. Diefenbacher, E. Eren, G. Kasieczka, A. Korol, B. Nachman, and D. Shih, DCTRGAN: Improving the precision of generative models with reweighting, J. Instrum. **15**, P11004 (2020).
[57] Benjamin Nachman and Jesse Thaler, Neural conditional reweighting, arXiv:2107.08979.
[58] Gilles Louppe, Michael Kagan, and Kyle Cranmer, Learning to pivot with adversarial networks, in *Advances in Neural Information Processing Systems*, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Curran Associates, Inc., 2017), Vol. 30.

[59] James Dolen, Philip Harris, Simone Marzani, Salvatore Rappoccio, and Nhan Tran, Thinking outside the ROCs: Designing Decorrelated Taggers (DDT) for jet substructure, J. High Energy Phys. 05 (2016) 156.
[60] Ian Moult, Benjamin Nachman, and Duff Neill, Convolved substructure: Analytically decorrelating jet substructure observables, J. High Energy Phys. 05 (2018) 002.
[61] Justin Stevens and Mike Williams, uBoost: A boosting method for producing uniform selection efficiencies from multivariate classifiers, J. Instrum. **8**, P12013 (2013).
[62] Chase Shimmin, Peter Sadowski, Pierre Baldi, Edison Weik, Daniel Whiteson, Edward Goul, and Andreas Søgaard, Decorrelated jet substructure tagging using adversarial neural networks, Phys. Rev. D **96**, 074034 (2017).
[63] Layne Bradshaw, Rashmish K. Mishra, Andrea Mitridate, and Bryan Ostdiek, Mass agnostic jet taggers, SciPost Phys. **8**, 011 (2020).
[64] 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.
[65] Gregor Kasieczka and David Shih, DisCo Fever: Robust Networks Through Distance Correlation, Phys. Rev. Lett. **125**, 122001 (2020).
[66] Li-Gang 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).
[67] Christoph Englert, Peter Galler, Philip Harris, and Michael Spannowsky, Machine learning uncertainties with adversarial neural networks, Eur. Phys. J. C **79**, 4 (2019).
[68] Stefan Wunsch, Simon Jórger, Roger Wolf, and Gunter Quast, Reducing the dependence of the neural network function to systematic uncertainties in the input space, Comput. Softw. Big Sci. **4**, 5 (2020).
[69] Alex Rogozhnikov, Aleksandar Bukva, V. V. Gligorov, Andrey Ustyuzhanin, and Mike Williams, New approaches for boosting to uniformity, J. Instrum. **10**, T03002 (2015).
[70] CMS Collaboration, A deep neural network to search for new long-lived particles decaying to jets, Mach. Learn. Sci. Technol. **1**, 035012 (2020).
[71] Jose M. Clavijo, Paul Glaysher, and Judith M. Katzy, Adversarial domain adaptation to reduce sample bias of a high energy physics classifier, Mach. Learn. Sci. Technol. **3**, 015014 (2022).
[72] Gregor Kasieczka, Benjamin Nachman, Matthew D. Schwartz, and David Shih, ABCDisCo: Automating the ABCD method with machine learning, Phys. Rev. D **103**, 035021 (2021).
[73] Ouail Kitouni, Benjamin Nachman, Constantin Weisser, and Mike Williams, Enhancing searches for resonances with machine learning and moment decomposition, J. High Energy Phys. 04 (2021) 070.
[74] Aishik Ghosh and Benjamin Nachman, A cautionary tale of decorrelating theory uncertainties, Eur. Phys. J. C **82**, 46 (2022).
[75] Andrew Blance, Michael Spannowsky, and Philip Waite, Adversarially-trained autoencoders for robust unsupervised new physics searches, J. High Energy Phys. 10 (2019) 047.
[76] Victor Estrade, Cécile Germain, Isabelle Guyon, and David Rousseau, Systematic aware learning—A case study in High Energy Physics, EPJ Web Conf. **214**, 06024 (2019).
[77] Stefan Wunsch, Simon Jörger, Roger Wolf, and Günter 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).
[78] 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).
[79] Pablo De Castro and Tommaso Dorigo, INFERNO: Inference-Aware Neural Optimisation, Comput. Phys. Commun. **244**, 170 (2019).
[80] Tom Charnock, Guilhem Lavaux, and Benjamin D. Wandelt, Automatic physical inference with information maximizing neural networks, Phys. Rev. D **97**, 083004 (2018).
[81] Justin Alsing and Benjamin Wandelt, Nuisance hardened data compression for fast likelihood-free inference, Mon. Not. R. Astron. Soc. **488**, 5093 (2019).
[82] Nathan Simpson and Lukas Heinrich, neos: End-to-end-optimised summary statistics for high energy physics, arXiv:2203.05570.
[83] Sven Bollweg, Manuel Haußmann, Gregor Kasieczka, Michel Luchmann, Tilman Plehn, and Jennifer Thompson, Deep-learning jets with uncertainties and more, SciPost Phys. **8**, 006 (2020).
[84] Jack Y. Araz and Michael Spannowsky, Combine and conquer: Event reconstruction with Bayesian Ensemble Neural Networks, J. High Energy Phys. 04 (2021) 296.
[85] Marco Bellagente, Manuel Haußmann, Michel Luchmann, and Tilman Plehn, Understanding event-generation networks via uncertainties, arXiv:2104.04543.
[86] Benjamin Nachman, A guide for deploying deep learning in LHC searches: How to achieve optimality and account for uncertainty, SciPost Phys. **8**, 090 (2020).
[87] Tommaso Dorigo and Pablo de Castro, Dealing with nuisance parameters using machine learning in high energy physics: A review, arXiv:2007.09121.
[88] Aishik Ghosh, Benjamin Nachman, and Daniel Whiteson, Uncertainty aware learning for high energy physics, Phys. Rev. D **104**, 056026 (2021).
[89] Fraciois Chollet, Keras, GitHub repository (2017), https://github.com/fchollet/keras.
[90] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard *et al.*, Tensorflow: A system for large-scale machine learning, in OSDI (USENIX Association, Savannah, Georgia, USA, 2016), Vol. **16**, pp. 265–283.
[91] Diederik Kingma and Jimmy Ba, Adam: A method for stochastic optimization, arXiv:1412.6980.
[92] Torbjorn Sjöstrand, Stephen Mrenna, and Peter Z. Skands, PYTHIA 6.4 physics and manual, J. High Energy Phys. 05 (2006) 026.
[93] Serguei Chatrchyan *et al.* (CMS Collaboration), Measurement of the underlying event activity at the LHC with

$\sqrt{s} = 7$ TeV and comparison with $\sqrt{s} = 0.9$ TeV, J. High Energy Phys. 09 (2011) 109.
[94] S. Agostinelli *et al.* (GEANT4 Collaboration), GEANT4–a simulation toolkit, Nucl. Instrum. Methods Phys. Res., Sect. A **506**, 250 (2003).
[95] J. Allison *et al.*, Geant4 developments and applications, IEEE Trans. Nucl. Sci. **53**, 270 (2006).
[96] J. Allison *et al.*, Recent developments in Geant4, Nucl. Instrum. Methods Phys. Res., Sect. A **835**, 186 (2016).
[97] S. Chatrchyan *et al.* (CMS Collaboration), The CMS experiment at the CERN LHC, J. Instrum. **3**, S08004 (2008).
[98] Torbjorn Sjöstrand, Stephen Mrenna, and Peter Z. Skands, A brief introduction to PYTHIA 8.1, Comput. Phys. Commun. **178**, 852 (2008).
[99] J. de Favereau, C. Delaere, P. Demin, A. Giammanco, V. Lemaître, A. Mertens, and M. Selvaggi (DELPHES 3 Collaboration), DELPHES 3, A modular framework for fast simulation of a generic collider experiment, J. High Energy Phys. 02 (2014) 057.
[100] Alexandre Mertens, New features in Delphes 3, J. Phys. Conf. Ser. **608**, 012045 (2015).
[101] Michele Selvaggi, DELPHES 3: A modular framework for fast-simulation of generic collider experiments, J. Phys. Conf. Ser. **523**, 012033 (2014).
[102] CMS Collaboration, Simulated dataset QCD_Pt-1000to1400_TuneZ2_7 TeV_pythia6 in AODSIM format for 2011 collision data (SM Exclusive), CERN Open Data Portal (2016), 10.7483/OPENDATA.CMS.96U2.3YAH.
[103] CMS Collaboration, Simulated dataset QCD_Pt-1400to1800_TuneZ2_7 TeV_pythia6 in AODSIM format for 2011 collision data (SM Exclusive), CERN Open Data Portal (2016), 10.7483/OPENDATA.CMS.RC9V.B5KX.
[104] CMS Collaboration, Simulated dataset QCD_Pt-1800_TuneZ2_7 TeV_pythia6 in AODSIM format for 2011 collision data (SM Exclusive), CERN Open Data Portal (2016), 10.7483/OPENDATA.CMS.CX2X.J3KW.
[105] Patrick T. Komiske, Radha Mastandrea, Eric M. Metodiev, Preksha Naik, and Jesse Thaler, Exploring the space of jets with CMS open data, Phys. Rev. D **101**, 034009 (2020).
[106] Patrick Komiske, Radha Mastandrea, Eric Metodiev, Preksha Naik, and Jesse Thaler, CMS 2011A Simulation | Pythia 6 QCD 1000-1400 | pT > 375 GeV | MOD HDF5 Format (2019).
[107] Patrick Komiske, Radha Mastandrea, Eric Metodiev, Preksha Naik, and Jesse Thaler, CMS 2011A Simulation | Pythia 6 QCD 1400-1800 | pT > 375 GeV | MOD HDF5 Format (2019).
[108] Patrick Komiske, Radha Mastandrea, Eric Metodiev, Preksha Naik, and Jesse Thaler, CMS 2011A Simulation | Pythia 6 QCD1800-inf | pT > 375 GeV | MOD HDF5 Format (2019).
[109] G. Kasieczka, B. Nachman, and D. Shih, Neural conditional reweighting, Phys. Rev. D **105**, 076015 (2022).
[110] Benjamin Nachman and Jesse Thaler, Delphes dijet dataset (2021).
[111] Matteo Cacciari, Gavin P. Salam, and Gregory Soyez, FastJet user manual, Eur. Phys. J. C **72**, 1896 (2012).
[112] Matteo Cacciari and Gavin P. Salam, Dispelling the N^3 myth for the k_t jet-finder, Phys. Lett. B **641**, 57 (2006).
[113] Matteo Cacciari, Gavin P. Salam, and Gregory Soyez, The anti-k_t jet clustering algorithm, J. High Energy Phys. 04 (2008) 063.
[114] Ouail Kitouni, Benjamin Nachman, Constantin Weisser, and Mike Williams, Enhancing searches for resonances with machine learning and moment decomposition, J. High Energy Phys. 04 (2021) 070.
[115] https://github.com/hep-lbdl/calibrationpriors.
[116] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, and Carol Willing, Jupyter notebooks—a publishing format for reproducible computational workflows, in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, edited by F. Loizides and B. Schmidt (IOS Press, 2016), pp. 87–90.
[117] Charles R. Harris *et al.*, Array programming with NumPy, Nature (London) **585**, 357 (2020).
[118] J. D. Hunter, Matplotlib: A 2d graphics environment, Comput. Sci. Eng. **9**, 90 (2007).
[119] http://iaifi.org/.