Moment extraction using an unfolding protocol without binning

Krish Desai,[1,2,*] Benjamin Nachman,[2,3,†] and Jesse Thaler[4,5,‡]

[1]*Department of Physics, University of California, Berkeley, California 94720, 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*
[4]*Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA*
[5]*The NSF AI Institute for Artificial Intelligence and Fundamental Interactions, USA*

(Received 30 July 2024; accepted 15 November 2024; published 13 December 2024)

Deconvolving ("unfolding") detector distortions is a critical step in the comparison of cross-section measurements with theoretical predictions in particle and nuclear physics. However, most existing approaches require histogram binning while many theoretical predictions are at the level of statistical moments. We develop a new approach to directly unfold distribution moments as a function of another observable without having to first discretize the data. Our moment unfolding technique uses machine learning and is inspired by Boltzmann weight factors and generative adversarial networks (GANs). We demonstrate the performance of this approach using jet substructure measurements in collider physics. With this illustrative example, we find that our moment unfolding protocol is more precise than bin-based approaches and is as or more precise than completely unbinned methods.

DOI: 10.1103/PhysRevD.110.116013

I. INTRODUCTION

Studying the dependence of physical observables on various quantities like energy scale offers a rich probe into the complex scaling dynamics of fundamental physical theories. In many cases, it is advantageous to summarize a probability distribution through a small number of statistical moments, which makes visualization and interpretation more tractable and lends itself to more precise theoretical predictions. For example, the spectra of many quark and gluon jet observables cannot be computed from first principles in perturbative quantum chromodynamics (QCD), but the energy dependence of their moments can be precisely predicted from factorization and Dokshitzer-Gribov-Lipatov-Altarelli-Parisi (DGLAP) evolution [1–3]. Additionally, one of the most precise extractions of the QCD coupling constant comes from comparing measured moments [4–15] to theoretical calculations [16].

Unfolding, also known as *deconvolution*, is the process of correcting detector distortions in experimental data. This is necessary for the accurate comparison of data between experiments, and with theoretical predictions. Typically, entire spectra are unfolded and then moments are computed afterward. In order to capture the dependence of an observable Z's moments on another quantity Y, these two features must be simultaneously unfolded. Current unfolding approaches discretize the (Z, Y) support, and then the two-dimensional histogram is unfolded such that the moments of Z can be computed in bins of Y. This binning procedure introduces discretization artifacts that hinder comparisons between measurements and theory, and between data from different experiments.

One possibility to improve the extraction of moments from data is to unfold without binning. A number of unbinned unfolding techniques have been proposed, including many based on machine learning [17–34] (see Refs. [35,36] for overviews). In terms of experimental applications, the OmniFold method [22,26] has recently been applied to studies of hadronic final states with data from H1 [37–40], LHCb [41], CMS [42,43], STAR [44], and ATLAS [45]. By construction, these unbinned approaches do not introduce binning artifacts. Nevertheless, because they offer a generic solution to unfolding entire spectra, unbinned methods may compromise precision for any particular aspect of the spectrum, such as a small set of moments. Furthermore, existing unbinned methods are also mostly iterative [22,26,28,34], which increases their computational complexity.

*Contact author: krish.desai@berkeley.edu
†Contact author: bpnachman@lbl.gov
‡Contact author: jthaler@mit.edu

In this paper, we introduce a dedicated machine learning-based unfolding method to directly unfold moments of observable distributions. Our moment unfolding technique is motivated by the Boltzmann distribution from statistical mechanics and uses a structure that is similar to generative adversarial networks (GANs) [46]. In particular, we learn a reweighting function at particle level whose form is determined by a Boltzmann weight factor so that its parameters can be identified with the observable moments. This function is optimized by requiring that the reweighted simulation at detector level is as similar as possible to the target data (determined by a discriminator), similar to the two-level GAN setups in Refs. [30,47,48]. Like OmniFold [22,26], our approach is based on reweighting, but it is fundamentally different because it is not iterative. We restrict our attention here to a small number of moments of a single observable. In principle, this approach could be extended to multiple observables and even full distributions, which we leave for future studies.

The remainder of this paper is organized as follows. We briefly review the statistics of moments in Sec. II and how these quantities are can be measured using binned or unbinned approaches. Our moment unfolding protocol is introduced in Sec. III. In Sec. IV, we provide numerical case studies, first on a Gaussian toy example and then on a realistic particle physics study involving jet substructure observables. The paper ends with conclusions and outlook in Sec. V.

II. THE STATISTICS OF MOMENT MEASUREMENTS

As a reminder, the kth moment of a probability density $p_Z(z)$ is calculated formally by taking an integral over the weighted density[1]

$$\langle Z^k\rangle \equiv \int_{-\infty}^{\infty} z^k p_Z(z)\mathrm{d}z. \tag{1}$$

When studying the dependence of the kth moment of Z on another observable Y, the probability density $p_Z(z)$ is replaced with the conditional probability density $p_{Z|Y}(z|y)$. Throughout this paper, we restrict our attention to the case that Z and Y are one-dimensional observables.

A. Biases from binning

To estimate the quantity in Eq. (1), one often uses a histogram approximation of $p_Z(z)$

$$\langle Z^k\rangle \approx \langle Z^k\rangle_{\text{bin}} \equiv \frac{1}{N}\sum_{i=1}^{n_{\text{bins}}} N_i z_{\text{bin},i}^k, \tag{2}$$

where N_i is the number of counts in bin i, $N=\sum_i N_i$ is the total number of counts, and $z_{\text{bin},i}$ is the center of bin i. In the limit that $N, n_{\text{bins}} \to \infty$ with equally spaced bins, $\langle Z^k\rangle_{\text{bin}} \to \langle Z^k\rangle$.[2]

To see that binning generically leads to biases, one can rewrite Eq. (1) as

$$\langle Z^k\rangle = \lim_{n_{\text{bins}}\to\infty}\sum_{i=1}^{n_{\text{bins}}}\int_{z_i}^{z_{i+1}} z^k p_Z(z)\mathrm{d}z \tag{3}$$

$$\equiv \lim_{n_{\text{bins}}\to\infty}\sum_{i=1}^{n_{\text{bins}}} p_i\langle Z^k\rangle_i, \tag{4}$$

where $\langle Z^k\rangle_i$ is the moment of z in bin i and p_i is the fraction of $p_Z(z)$ that falls in bin i. Therefore, in the limit that $N\to\infty$, but n_{bins} is finite, the bias due to the binning is

$$\langle Z^k\rangle - \langle Z^k\rangle_{\text{bin}} = \sum_{i=1}^{n_{\text{bins}}}(\langle Z^k\rangle_i - z_{\text{bin},i}^k). \tag{5}$$

This equation emphasizes that, if instead of using the bin centers as in Eq. (2), one were to use the k^{th} moment per bin, then the binning bias could be removed. The histogramming tool YODA [49] keeps track of first and second moments within bins for precisely this reason.

For spectra that are monotonically increasing or decreasing, Eq. (5) predicts the sign of the bias. For spectra that have one or more maxima, it is not possible to even know, in general, if there is a bias and if so, what is the sign of the bias.

B. Unfolding binned measurements

In an experimental context, before computing $\langle Z^k\rangle_{\text{bin}}$ in Eq. (2), it is necessary to estimate N_i. A variety of regularized matrix inversion approaches have been proposed, which use a response matrix $\mathbf{R}$ to relate the counts at detector level to the counts at particle level. In particular, the folding equation can be written as $\mathbf{x}=\mathbf{R}\mathbf{z}$, where $\mathbf{x}$ and $\mathbf{z}$ are vectors with the detector-level and particle-level counts, respectively, and $\mathbf{R}$ is the response matrix. The elements of the response matrix are

$$R_{ij} = \Pr(\text{measure in bin } i|\text{truth is bin } j), \tag{6}$$

where $\Pr(\cdot)$ indicates probability of the argument. Note that in general, $\mathbf{R}$ need not be a square matrix, which is one reason why simple matrix inversion is not typically effective for unfolding.

The most common approaches to inferring $\mathbf{z}$ from $\mathbf{x}$ include iterative Bayesian unfolding (IBU) [50] (also known as Richardson-Lucy deconvolution [51,52]), singular value decomposition (SVD) [53], and TUnfold [54].

[1]Upper-case letters represent random variables and lower-case letters represent realizations of those random variables.

[2]Often nonuniform bin spacing is used to accommodate nonlinear detector resolutions. The statement in the text is true more generally when the maximum bin width goes to zero.

Reviews on unfolding methods can be found in Refs. [55–58]. The method we will use as a baseline for the case studies in Sec. IV is IBU, which proceeds iteratively

$$z_j^{(t)} = \sum_i \Pr^{(t-1)}(\text{truth is } j|\text{measure } i)\Pr(\text{measure } i) = \sum_i \frac{R_{ij} z_j^{(t-1)}}{\sum_m R_{im} z_m^{(t-1)}} \times x_i, \tag{7}$$

where $z^{(0)}$ is a prior distribution (often taken to be the particle-level simulation) and t is the iteration number.

After unfolding, a number of classical approaches have been proposed to correct for the bias in Eq. (5). Perhaps the most common is to apply a multiplicative correction to the binned unfolded data

$$\langle Z^k \rangle_{\text{meas}} = \langle X^k \rangle_{\text{bin,data}} \times \frac{\langle Z^k \rangle_{\text{MC truth}}}{\langle X^k \rangle_{\text{bin,MC reco}}}, \tag{8}$$

where $\langle Z^k \rangle_{\text{MC truth}}$ is the moment computed in simulation without binning. A challenge with this approach is that it does not make use of any local information in Z, since all values that enter in the above equation are summed over bins. To solve this, one could apply a correction per Z bin

$$\langle Z^k \rangle_{\text{meas}} = \frac{1}{N} \sum_{i=1}^{n_{\text{bins}}} N_i \langle Z^k \rangle_{\text{MC truth},i}, \tag{9}$$

where $\langle Z^k \rangle_{\text{MC truth},i}$ is the mean value of X in the ith bin from simulation. In this case, one relies on the prior density within a given bin of Z, but if the prior is not too different from nature, the resulting bias will be suppressed. It may be possible to further improve this by using data to unfold also the values of $\langle Z^k \rangle_i$.

C. Unbinned unfolding methods

One way to completely avoid the bias in Eq. (5) is to unfold without binning in the first place. There are a number of unbinned unfolding methods, but to our knowledge, the only one applied to data so far is OmniFold [22,26], which generalizes IBU. It uses neural network classifiers to iteratively reweight the particle- and detector-level Monte Carlo events, respectively. The final product of OmniFold is a weighting function $\nu(z)$ so that the unfolded expectation value of any observable computable from Z is given by

$$\langle \mathcal{O} \rangle = \sum_{z \in \text{Gen}} \nu(z) \mathcal{O}(z), \tag{10}$$

where the sum runs over synthetic particle-level events from a Monte Carlo generator. Usually, $\mathcal{O}$ represents the counts in a given bin of a histogram from a differential cross section measurement. However, $\mathcal{O}$ could also be the k^{th} moment of Z directly. A key benefit of OmniFold is that all unfolded expectation values are derived simultaneously from a single reweighting function.

Because of its generality, OmniFold may not be as precise for any particular observable and moment. In principle, OmniFold is capable of unfolding all observables and all moments simultaneously and studies have shown that adding more features can improve the precision on a given observable [22]. However, the same studies have also shown that adding more information can reduce precision. This may be due to cases where the gains from new information covariate with the detector response are outweighed by the additional regularization needed to fit higher-dimensional data. Detailed studies of this bias-variance trade off would be interesting to explore in the future.

A computational challenge with OmniFold and most other methods that actively mitigate prior dependence [22,26,28,34] is that they are iterative. In practice, this means that unfolding may require training tens of neural networks (one for each step of each iteration), which can easily reach thousands when ensembling is added into the workflow to achieve stability.

III. MOMENT UNFOLDING

Motivated by the above challenges, we introduce moment unfolding, which can directly learn moments without first unfolding the entire spectrum. Moment unfolding is an unbinned, noniterative, reweighting-based method to unfold the statistical moments of observables, inspired by Boltzmann's approach to construct the Maxwell–Boltzmann distribution [59].

For the following discussion, we use the nomenclature of Ref. [22], where unfolding involves four datasets: truth, data, generation, and simulation. Each synthetic collision event comes as a pair (Z, X), for $Z \in \mathbb{R}^{N_{\text{Gen}}}$ and $X \in \mathbb{R}^{N_{\text{Sim}}}$, where Z is the predetector version of the event ("generation") and X is the postdetector observation of the event ("simulation"). In experimental data, we only have access to the detector-level version ("data"), so we use the simulation to infer the underlying predetector distribution ("truth").

A. Leveraging Boltzmann weights

Moment unfolding uses a weight function $g(z)$, similar to OmniFold's $\nu(z)$ in Sec. II C. Instead of determining the weight functions in an iterative fashion, though, moment unfolding uses a fixed functional form

$$g(z) = \frac{1}{P} \exp\left[-\sum_{a=1}^{n} \beta_a z^a \right], \tag{11}$$

where n is the number of moments to be simultaneously unfolded, β_a are parameters to be determined, and P is a normalization constant, similar to the partition function from statistical mechanics. When we want to unfold moments conditional on another observable y, the parameters β_a are replaced with functions $\beta_a(y)$, as discussed more in Sec. IV C. The exponential form of Eq. (11) is inspired by the $e^{-\beta E}$ Boltzmann factor from statistical physics, whose derivation is reviewed in Appendix A.

To better understand this choice, recall that the Maxwell–Boltzmann distribution is the one that maximizes the entropy of an ensemble while holding mean energy constant. This logic can be extended beyond means to arbitrary constraints, yielding the maximum entropy probability distribution [59]. In this language, Eq. (11) optimizes the relative entropy of the reweighted truth-level distribution with respect to the Z prior, while holding the first n moments fixed to some value.

As described in more detail in Sec. III B, we determine the values of β_a by maximizing the maximum likelihood classifier loss [60–62] between the reweighted detector-level simulation and experimental data.[3] Crucially, the learned values of β_a are *not* the learned moments themselves. Rather, analogously to Eq. (10), the moments are given by

$$\langle Z^k \rangle_{\text{Moment Unfolding}} = \sum_{z \in \text{Gen}} g(z) z^k, \tag{12}$$

where the sums run over synthetic particle-level events, and the normalization P is determined numerically

$$P = \sum_{z \in \text{Gen}} \exp\left[-\sum_a \beta_a z^a\right]. \tag{13}$$

In this way, the extracted k^{th} moment depends on all n β_a values.[4]

There is some arbitrariness in the choice of $g(z)$, since many weight functions with n free parameters can in principle be used to match n moments of a distribution. The advantage of our choice of $g(z)$ is that, for the training procedure described below, moment unfolding provably converges to the truth moments under certain conditions, as described in Appendix B.

The hyperparameter n sets the degree of the polynomial in the exponent, i.e., the number trainable weights in the generator and consequently the number of moments that are unfolded. One might attempt to perform this procedure for arbitrarily large values of n to reconstruct arbitrarily high moments. However, n also simultaneously serves as a regularization parameter that restricts the class of generator functions that the algorithm can optimize over. As $n \to \infty$, this class is the set of all positive analytic functions. This is a manifestation of the bias-variance trade-off; increasing n to reconstruct higher moments results in a reduction in the precision of the prediction of any individual moment.

[3]See Ref. [63] for related discussions using the binary cross entropy loss.

[4]This distinction is why we use k to index the measured moments but a to index the learned parameters.

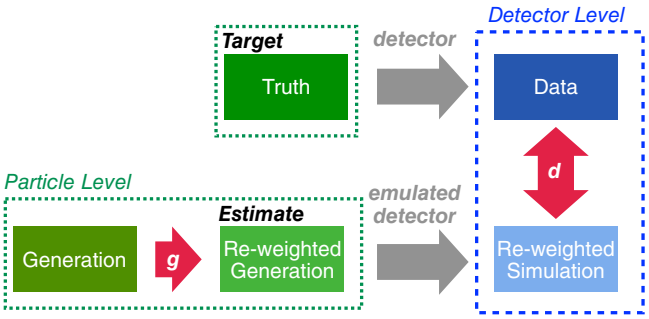

FIG. 1. A schematic diagram of the training setup for moment unfolding. Like a GAN, g is the generator and d is the discriminator, but now g is simply a reweighting factor given by Eq. (11). The reweighted simulation dataset inherits its weight from the matching generation dataset. The detector emulations are only run once, since a new simulated dataset is created via importance weights and not by changing the features themselves.

B. Adversarial optimization

To implement moment unfolding, we modify the learning setup of a generative adversarial network (GAN) [46] to find the optimal values of β_a in Eq. (11). As shown schematically in Fig. 1, the weight function $g(z)$ can be viewed as a "generator" which is optimized adversarially against a "discriminator" that tries to distinguish the reweighted simulation from the experimental data.

In a typical GAN, the generator g surjects a latent space onto a data space, while a discriminator d distinguishes generated examples from real examples. These two neural networks are then trained simultaneously to optimize the binary cross entropy (BCE) loss functional, where the generator tries to maximize the loss with respect to g while the discriminator tries to minimize the loss with respect to d.

For moment unfolding, the latent space probability density is the truth-level simulation density, and the generation process is simply reweighting events by $g(z)$, where $g(z) = g_{\text{NN}}(z)/\hat{P}$ for neural network g_{NN} and batch-level normalization estimate $\hat{P}$. Our discriminator is a neutral network $d(x)$ that operates on detector-level distributions. Instead of BCE, we use the maximum likelihood classifier (MLC) [60–62] loss functional, because it satisfies the analytic closure guarantees proven in Appendix B. That said, we tested BCE for our case studies, finding that it yields similar empirical performance. The functions $g(z)$ and $d(x)$ are trained simultaneously to

optimize a weighted version of MLC loss

$$L[g,d] = -\sum_{x\in\text{Data}} \log d(x) - \sum_{(z,x)\in(\text{Gen,Sim})} g(z)(1-d(x)), \tag{14}$$

In the first sum, x are examples obtained from data, while in the second sum, (z,x) are tuples sampled from generation and simulation. A similar setup (with unrestricted g) was used for domain adaptation in Ref. [64].

For the empirical studies in Sec. IV, each neural network is implemented using Keras [65] with the TensorFlow2 backend [66] and optimized with ADAM [67]. The discriminator function d has three hidden layers, using 50 nodes per layer. Rectified Linear Unit (ReLU) activation functions are used for the intermediate layers and a sigmoid function is used for the last layer.

IV. CASE STUDIES

To demonstrate the features of moment unfolding, we perform numerical cases studies, both in a Gaussian example and in the realistic setting of jet measurements at the Large Hadron Collider (LHC).

A. Gaussian example

We begin by unfolding Gaussian data with Gaussian distortions. Let $\mathcal{N}(\mu,\sigma^2)$ be a normal distribution with mean μ and variance σ^2. The particle-level truth distribution is drawn from $\mathcal{N}(0,1)$ while the particle-level generation is drawn from $\mathcal{N}(-0.5,1)$. Detector effects are represented by additive noise that is also Gaussian, with distribution $\mathcal{N}(0,5)$. Since the Gaussian probability density is uniquely specified by its first two moments, moment unfolding with weighting function

$$g(z) = \frac{1}{P} e^{-\beta_1 z - \beta_2 z^2} \tag{15}$$

can in principle result in a perfect unfolding of the entire distribution. As discussed in Appendix B, no unfolding method can be successful in all cases, and moment unfolding in particular cannot recover the true distribution when there are large off-diagonal elements in the detector response, which is indeed the case here.

Nevertheless, as shown in Fig. 2(a), the numerical results of applying moment unfolding to this Gaussian example are quite promising. Here, we show histograms at particle level, comparing the truth dataset (blue shaded), generation dataset (orange shaded), and the result of weighting the generation dataset by Eq. (15) (black dotted line). Visually, the close overlap between the truth histogram and the weighted generation histogram shows that the moment unfolding procedure was successful.

Since this is a simple one-dimensional problem, we can study the success of this procedure more precisely. Specifically, we can check whether the maximum of the discriminator loss function is indeed at values of β_a that correspond to the moments of the truth distribution. In practice, we do not have access to the full loss landscape, but for this one-dimensional problem, we can scan over a discrete set of generator parameters for illustration. Then, to obtain the value of the loss, we can train the discriminator for fixed values of β_a.

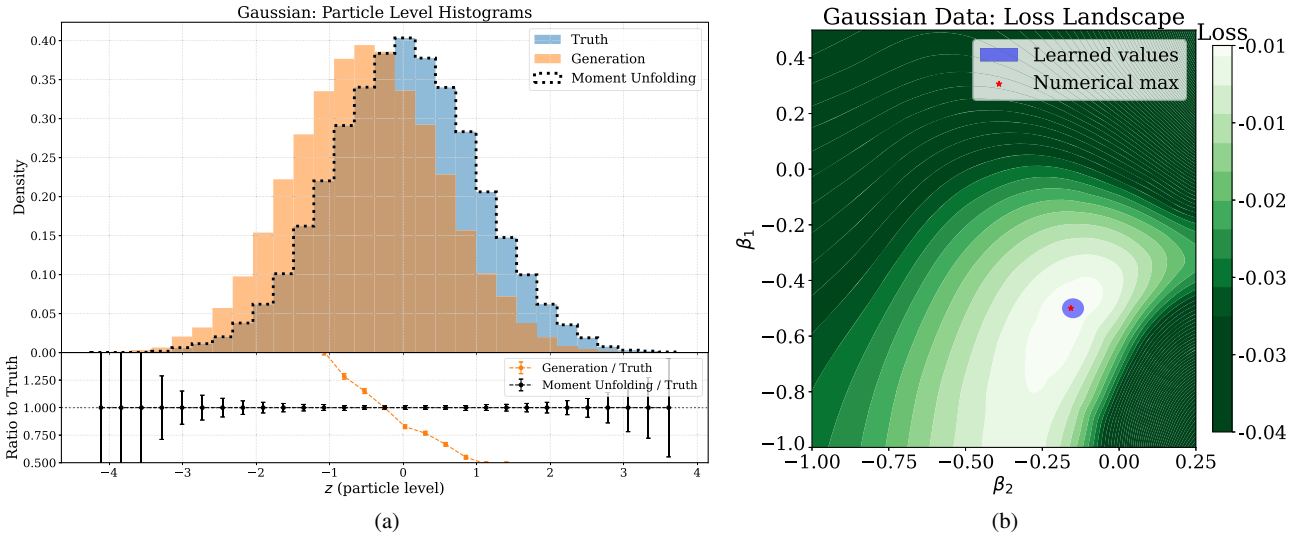

FIG. 2. (a) Distributions from the Gaussian example of particle-level truth, generation, and reweighted generation (i.e., moment unfolding). The agreement between the truth and reweighted samples demonstrates the qualitative performance of moment unfolding. (b) The weighted MLC loss from Eq. (14) for fixed g but optimized d, found by scanning over β_1 and β_2. The correct value is indicated by a red star. Indicated in shaded blue is the 1σ bootstrapped interval for moment unfolding's prediction of β_a.

In Fig. 2(b), we plot the discriminator-optimized loss as a function of β_1 and β_2. The red star represents the loss maximum. The solid blue region represents the 1σ confidence interval for the values of β_a learned by the moment unfolding algorithm, estimated from a bootstrapping procedure. Since the red star coincides with the corresponding solid blue ellipse, the success of the procedure is verified.

B. Jet substructure

As a more realistic case study, we study hadronic jets from the LHC. Jets are collimated sprays of particles that arise from the fragmentation of high-energy quarks and gluons. Measuring the substructure of jets is an active area of research, both to understand QCD dynamics and to search for physics beyond the Standard Model [68,69]. For this study, we consider four jet substructure observables: jet mass m, jet charge q with $\kappa = 1/2$ [70], jet width w [68], and momentum fraction z_g [71] after Soft Drop grooming [72,73] with $z_{\text{cut}} = 0.1$ and $\beta = 0$:

$$m = \sqrt{\sum_k E_k^2 - \sum_k p_k^2}, \tag{16}$$

$$q = \frac{1}{\sum_k \sqrt{p_{T,k}}} \sum_k q_k \sqrt{p_{T,k}}, \tag{17}$$

$$w = \frac{1}{p_{T,\text{jet}}} \sum_k p_{T,k} \Delta R_k, \tag{18}$$

$$z_g = \frac{p_{T,\text{subleading}}}{p_{T,\text{leading}} + p_{T,\text{subleading}}}. \tag{19}$$

Here, the sums run over the constituents of a jet, and E_k, p_k, $p_{T,k}$, ΔR_k, and q_k are the constituent energy, three-momentum, transverse momentum, angular distance from the jet axis, and electric charge for particle k, respectively, and $p_{T,(\text{sub})\text{leading}}$ is the transverse momentum of the (sub)leading prong [74] returned by the Soft Drop algorithm. The jet mass and jet width are examples of infrared-and-collinear-safe observables that are expected to be less sensitive to detector effects, while the groomed momentum fraction (as $\beta \to 0$) is Sudakov safe [71]. Jet charge is infrared but not collinear safe and therefore expected to be more susceptible to certain kinds of detector distortions. The first moment of the jet charge distribution as a function of jet p_T has been calculated in Refs. [70,75–78] and measured by ATLAS [79] and CMS [80,81].

The simulated samples used for this study are the same as in Refs. [22,82] and briefly summarized here. Proton-proton collisions are simulated at $\sqrt{s} = 14$ TeV with the default tune of Herwig 7.1.5 [83–85] and Tune 26 [86] of Pythia 8.243 [87–89]. As a proxy for detector effects and a full detector simulation, we use the DELPHES 3.4.2 [90] fast simulation of the CMS detector, which uses particle flow reconstruction [91]. Jets with radius parameter $R = 0.4$ are clustered using either all particle flow objects (detector-level) or stable non-neutrino truth particles (particle-level) with the anti-k_T algorithm [92] implemented in FastJet 3.3.2 [93,94]. To reduce acceptance effects, the leading jets are studied in events with a Z boson with transverse momentum $p_T^Z > 200$ GeV. For this study, we treat Pythia + DELPHES as "truth/data" and Herwig + DELPHES as "generation/simulation."

For each of the four observables above, we unfold the first two moments from detector-distorted jet substructure data. We emphasize that this unfolding is done separately for each observable, leaving joint unfolding to future work. The moment results are presented in Table I, where we see that moment unfolding has good performance recovering the expected truth moments within statistical uncertainties. The uncertainties in the "truth" and "generation" columns are computed by bootstrapping the respective datasets, computing the relevant moment for each bootstrapped dataset, and then computing the 1σ interval for this moment. The uncertainties in the "moment unfolding" column are computed by adding the uncertainty in

TABLE I. Moments of jet observables at particle level. The uncertainties in the truth and generation columns are computed by bootstrapping the datasets. The uncertainties in the moment unfolding column are computed by adding the uncertainty in Generation in quadrature to the empirical uncertainty obtained by computing the 1σ confidence interval for the moment predicted by moment unfolding on the same dataset multiple times.

Observable	Truth	Generation	Moment unfolding
$\langle M \rangle$	$(2.182 \pm 0.030) \times 10^{1}$	$(2.064 \pm 0.043) \times 10^{1}$	$(2.173 \pm 0.047) \times 10^{1}$
$\langle M^2 \rangle$	$(6.049 \pm 0.222) \times 10^{2}$	$(5.360 \pm 0.350) \times 10^{2}$	$(6.115 \pm 0.364) \times 10^{2}$
$\langle Q \rangle$	$(1.006 \pm 0.037) \times 10^{-2}$	$(1.582 \pm 0.038) \times 10^{-2}$	$(1.090 \pm 0.040) \times 10^{-2}$
$\langle Q^2 \rangle$	$(1.216 \pm 0.082) \times 10^{-2}$	$(1.508 \pm 0.074) \times 10^{-2}$	$(1.207 \pm 0.074) \times 10^{-2}$
$\langle W \rangle$	$(1.498 \pm 0.025) \times 10^{-1}$	$(1.231 \pm 0.029) \times 10^{-1}$	$(1.499 \pm 0.029) \times 10^{-1}$
$\langle W^2 \rangle$	$(3.370 \pm 0.113) \times 10^{-2}$	$(2.421 \pm 0.128) \times 10^{-2}$	$(3.374 \pm 0.128) \times 10^{-2}$
$\langle Z_g \rangle$	$(2.334 \pm 0.029) \times 10^{-1}$	$(2.457 \pm 0.030) \times 10^{-1}$	$(2.353 \pm 0.059) \times 10^{-1}$
$\langle Z_g^2 \rangle$	$(6.789 \pm 0.166) \times 10^{-2}$	$(7.425 \pm 0.165) \times 10^{-2}$	$(6.767 \pm 0.330) \times 10^{-2}$

"generation" in quadrature to the intrinsic uncertainty of the empirical procedure estimated by learning the same moment on the same dataset multiple times and computing the 1σ confidence interval for the predicted moment.

As an alternative visualization, we show the inferred particle-level distributions in Fig. 3. The jet mass follows a unimodal distribution peaked at about 20 GeV, with a relatively long tail many standard deviations to the right. These features are observed both in truth and generation with a sharper peak in truth. The jet charge has an approximately Gaussian distribution, and is close to symmetric with a small positive skew because approximately equal number of positively and negatively charged particles are produced, with a small excess of positively charged particles produced because this is a proton-proton process. Radiation within jets is enhanced at low values of ΔR which leads to the unimodal distribution of the jet width that falls off rapidly after about 0.1; these features are present in both the truth and generation, albeit with a longer tail in truth. The groomed momentum fraction offers an opportunity to study the performance of moment unfolding for data that is not well approximated as a Gaussian distribution and has a sharp cutoff feature. For all four observables, even though the first and second moments of the reweighted generation match the truth well, the full distributions are not statistically identical. This is because higher moments are relevant and are not the same between truth and generation.

In Fig. 4, we perform a loss function analysis where we scan over values of β_a, learn the optimal discriminator for the fixed generator, and then compare the loss function maximum to the learned values of β_a. The red star represents the maximum of the MLC loss. The solid blue region represents the 1σ confidence interval for the values of β_a learned by the moment unfolding algorithm,

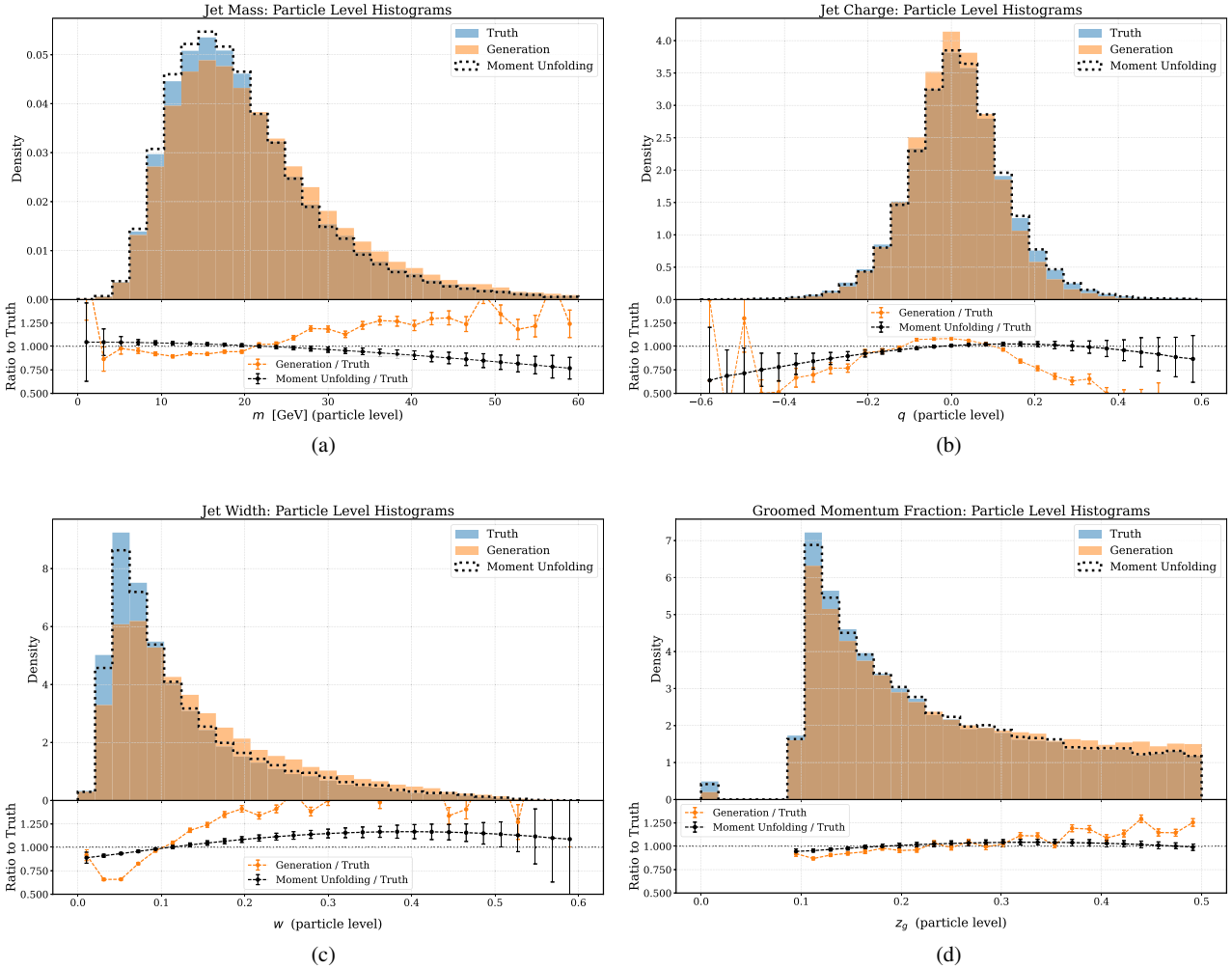

FIG. 3. Distributions of (a) jet mass, (b) jet charge, (c) jet width, and (d) groomed momentum fraction at particle-level, comparing truth, generation, and the results from moment unfolding.

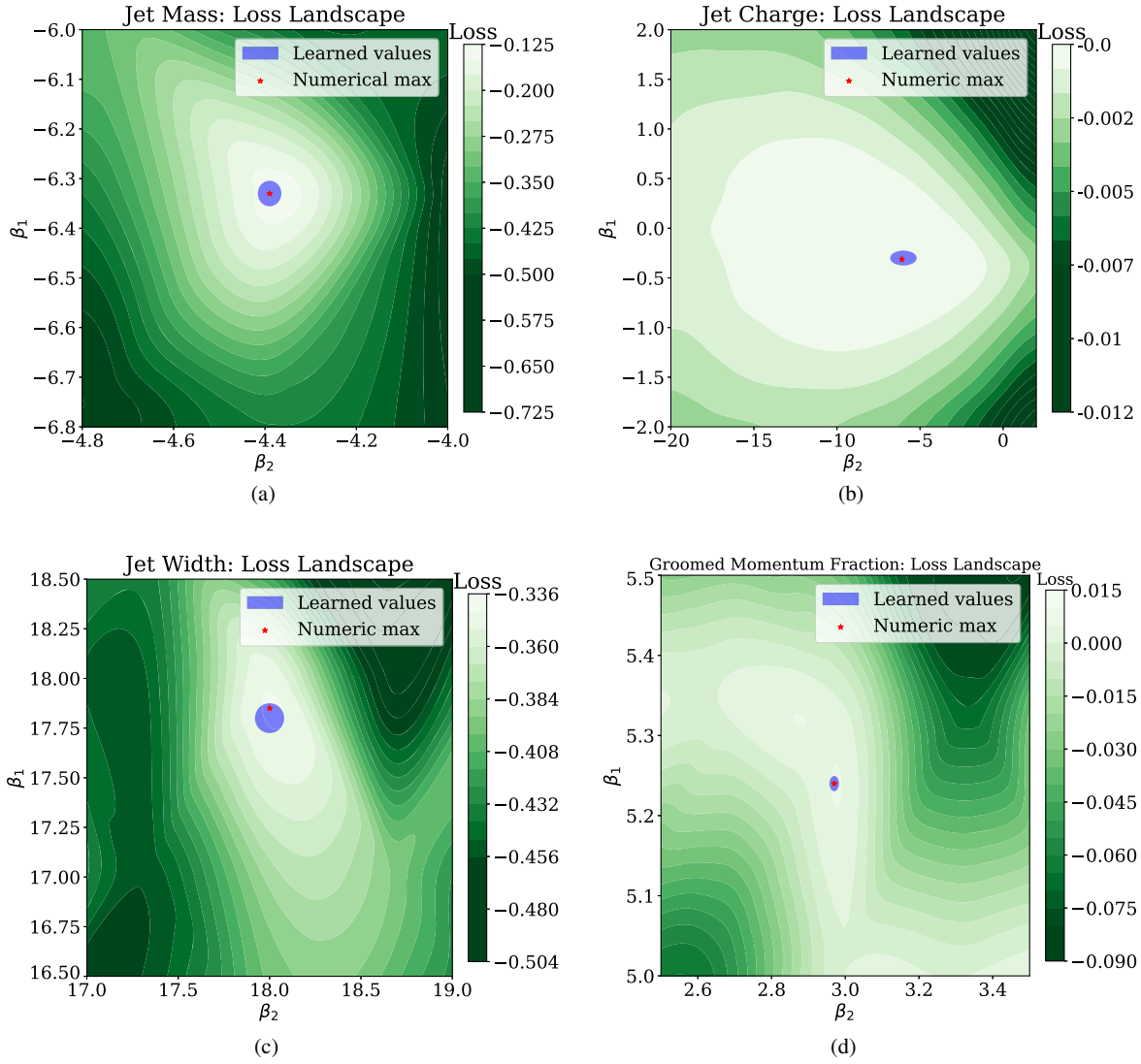

FIG. 4. The discriminator-optimized MLC loss as a function of β_1 and β_2, for (a) jet mass, (b) jet charge, (c) jet width, and (d) groomed momentum fraction. The correct β_a values are indicated by red dot, while the 1σ intervals for moment unfolding's predictions of β_a are shown as a blue circle.

estimated from a bootstrapping procedure. Since the red star coincides with the corresponding solid blue ellipse, the success of the procedure is verified.

C. Momentum dependence

Typically, we are interested in more than just inclusive moments. For jet substructure observables, it is interesting to study the moments as a function of jet p_T. We can slightly modify our generator g to accommodate this case by adding momentum dependence to the coefficients in Eq. (11)

$$g(z;p_T) = \frac{1}{P}\exp\left[-\sum_{a=1}^{n}\beta_a(p_T)z^a\right], \tag{20}$$

where $\beta_a(p_T)$ is an arbitrary function of p_T. While we could parametrize $\beta_a(p_T)$ as a neural network, we found that this resulted in unstable performance. Since the p_T-dependence is often weak and since the starting simulations are already quite accurate, we regularize the β_a by parametrizing them as low-order polynomials. Empirically, we observe that the ratio of spectra between Pythia and Herwig is approximately linear, so we restrict the β_a to be first-order polynomials in p_T

$$\beta_a(p_T) = \beta_a^{(0)} + \beta_a^{(1)}p_T. \tag{21}$$

In Fig. 5, we show the results of carrying out this procedure for the jet mass, jet charge, jet width, and groomed momentum fraction on the inclusive distribution.

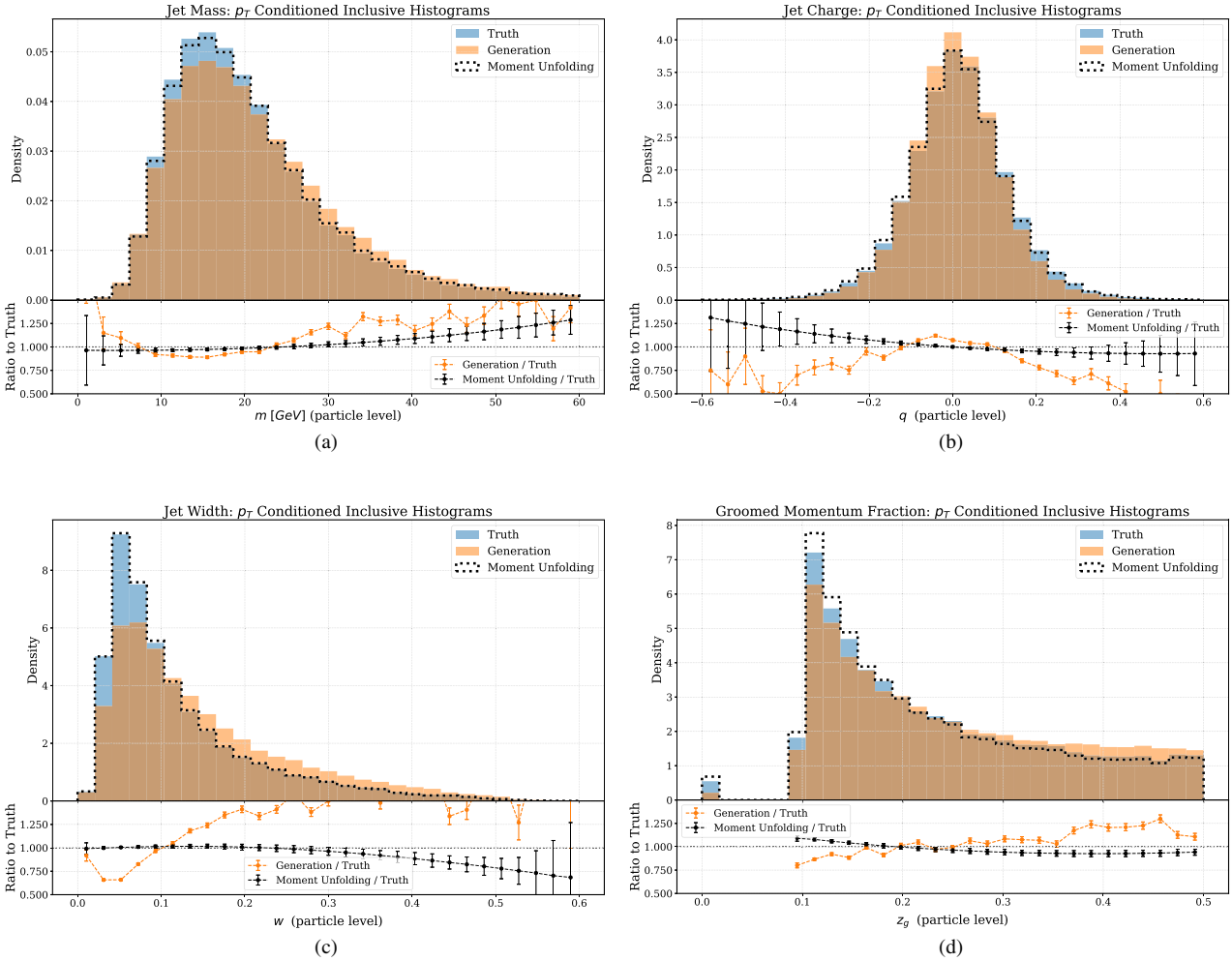

FIG. 5. Inclusive distributions of (a) jet mass, (b) jet charge, (c) jet width, and (d) groomed momentum fraction unfolded conditionally on the traverse momentum of the jet. Compared to the unconditional unfolding in Fig. 3, the agreement between truth and moment unfolding is typically better.

These can be contrasted with the plots in Fig. 3 which did not have p_T dependence in the unfolding. With a p_T-dependent weight function, there is stronger similarity between the truth and moment unfolding. Having observed the improved inclusive behavior, we now study jet observable moments differentially in p_T.

In Fig. 6, we plot the dependence of jet observable moments on the jet p_T. The left column shows the first moments of the jet mass, jet charge, and jet width, while the right column shows the corresponding second moments. The moments are computed in bins of transverse momentum for the truth dataset (blue triangles), the generation dataset (orange triangles), and moment unfolding result using the weight factor $g(x; p_T)$ from Eq. (20) (black circles). Up to statistical uncertainties, we see that the moment unfolding results coincide with the moments of the truth dataset, which are substantially distinct from the moments of the generation dataset.

D. Comparison to other methods

Finally, we compare the unfolded moments computed through moment unfolding method to those obtained through three alternative unfolding methods:

(i) *OmniFold*: An example of unbinned unfolding.
(ii) *IBU*: An example of binned unfolding.
(iii) *IBU + Bin Correction*: Same as above but performing the binwise correction from Eq. (9).

The results of this comparison are shown in Fig. 7, for the first and second moments of jet mass, jet charge, jet width, and groomed momentum fraction. The top panel of each plot shows the moments as a function of jet p_T, comparing moment unfolding (black circles), the three method listed above (IBU in green squares, IBU with the binwise correction in yellow diamonds, and Omnifold in red triangles), and the truth dataset (blue triangles). To better highlight the performance of each method, the bottom panels of each plot show the ratio of the unfolded moments to the truth moments.

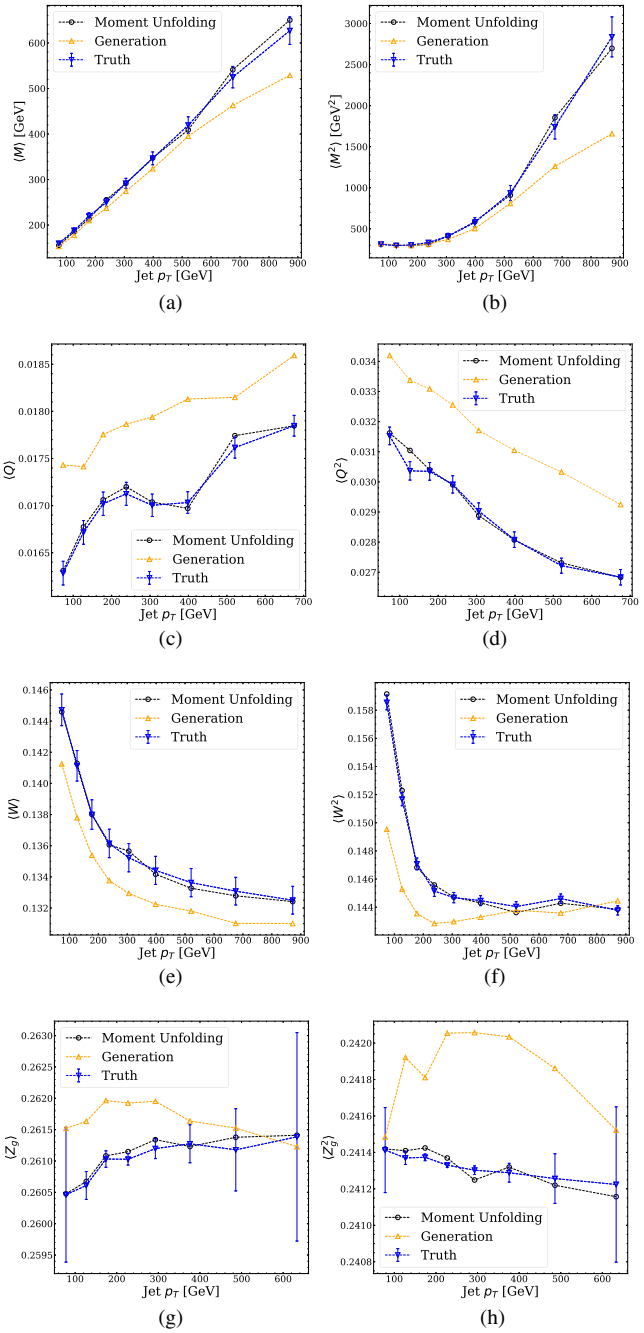

FIG. 6. The mean (left column) and variance (right column) of the jet mass (a, b), jet charge (c, d), jet width (e, f), and groomed momentum fraction (g, h) as a function of the transverse momentum of the jet. For visual clarity, only statistical uncertainties on the truth distribution are shown. With uncertainties, the moment unfolding results are in good agreement with the truth.

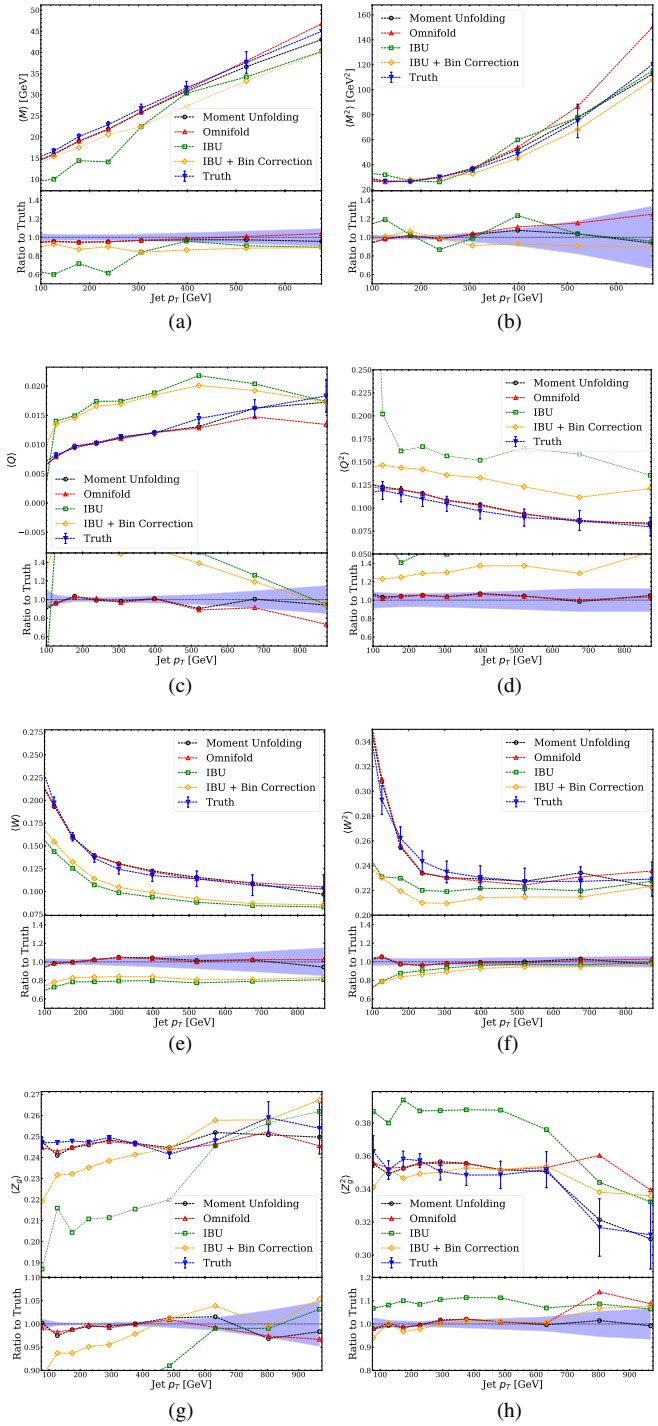

FIG. 7. Same observables and moments as Fig. 6, but now comparing the moment unfolding result to those of OmniFold, IBU, and IBU with binwise correction. The lower panel in each plot shows the ratio of the extracted moments to the truth as a function of jet p_T, demonstrating the strong performance of moment unfolding relative to IBU, and comparable performance to OmniFold.

In general, the unbinned methods OmniFold and moment unfolding outperform both versions of IBU. Despite having a more rigid reweighting function and simpler training paradigm, moment unfolding nevertheless exhibits comparable (and in some cases) better performance than OmniFold. On average, moment unfolding takes about 10^4 times longer to run than IBU, and Omnifold takes about 10^2 times longer to run than moment unfolding.

V. CONCLUSIONS AND OUTLOOK

In this paper, we introduced the first dedicated approach to unfolding the moments of distributions without binning. Our moment unfolding protocol is based on the structure of a generative adversarial network, where the generator is a weighting function at particle level and the discriminator is a classifier acting at detector level. The weight function is inspired by the Boltzmann distribution, with a small number of parameters that have a physical interpretation as Lagrange multipliers imposing moment constraints.

Through both a simple Gaussian example and physically relevant examples from jet physics, we showed that moment unfolding is able to recover the desired truth moments. The performance is comparable to a generic approach for unbinned unfolding (OmniFold), but without the complexity of an iterative algorithm. Moment unfolding is able to recover moments inclusively, and with a small modification, also differential in at least one quantity. While the dependence of one moment on one other observable is a common case, this new method can in principle be extended to more moments and differential in more quantities, though the practical challenges of this scaling is left to future studies.

Going to the extreme limit, the moment unfolding strategy could be extended to full distributions. Unfolding full distributions typically requires some kind of regularization, such as limiting the number of iterations when using iterative methods. For a noniterative method like moment unfolding, one would need to regularize the functional form of the weight factor in some way, for example by using neural networks with a Lipschitz constraint [95,96]. With appropriate regularization, a generalized version of moment unfolding could potentially combine the flexibility of machine-learning-based approaches like OmniFold with the robustness of traditional unfolding strategies.

ACKNOWLEDGMENTS

We thank Shuchin Aeron, Benjamin Fischer, and Dennis Noll for useful discussions. B. N. would like to thank Stefan Kluth for discussions about unfolding moments nearly a decade ago in the context of Ref. [97]. J. T. would like to thank Benoit Assi, Stefan Hoeche, and Kyle Lee for discussions of moments in the context of theory calculations. K. D. and B. N. are supported by the U.S. Department of Energy (DOE), Office of Science under Contract No. DE-AC02-05CH11231. J. T. is supported by the National Science Foundation (NSF) under Cooperative Agreement No. PHY-2019786 (The NSF AI Institute for Artificial Intelligence and Fundamental Interactions, [98]), by the U.S. DOE Office of High Energy Physics under Grant No. DE-SC0012567, by the Simons Foundation through Investigator Grant No. 929241, and his work was performed in part at the Aspen Center for Physics, which is supported by NSF Grant No. PHY-2210452.

DATA AVAILABILITY

The code for this paper can be found at [100], which makes use of Jupyter notebooks [99] employing NumPy [101] for data manipulation and Matplotlib [102] to produce figures. All of the machine learning was performed on an Nvidia RTX6000 Graphical Processing Unit (GPU) and running the notebook to perform unfold the first few moments of a dataset takes less than five minutes per iteration (to extract bootstrapped uncertainties we perform 500 iterations). The physics data sets are hosted on Zenodo at [22,82].

APPENDIX A: REVIEW OF BOLTZMANN WEIGHTS

In this appendix, we derive the weight factor from Eq. (11), following the familiar derivation of the Boltzmann distribution. The goal is to learn a distribution $\ell(z)$ that optimizes the relative entropy of $\ell(z)$ with respect to a prior $q(z)$, subject to moment constraints from a distribution $p(z)$.

The KL divergence of $\ell(z)$ from $q(z)$ is

$$D_{\mathrm{KL}}(\ell\|q) = \int \ell(z) \log \frac{\ell(z)}{q(z)} \mathrm{d}z. \tag{A1}$$

In the absence of constraints, this quantity would be minimized when $\ell(z) = q(z)$. Note that the KL divergence is not symmetric between $\ell(z)$ and $q(z)$, which is essential to the following derivation.

To impose the moment constraints, we include Lagrange multipliers β_a to force $\ell(z)$ to have the same first n moments as $p(z)$. This corresponds to the loss function

$$L = D_{\mathrm{KL}}(\ell\|q) + \sum_{a=0}^{n} \beta_a \int z^a (\ell(z) - p(z)) \mathrm{d}z, \tag{A2}$$

where the $a = 0$ term enforces that $\ell(z)$ is properly normalized. Taking a functional derivative of the loss with respect to $\ell(z)$

$$\frac{\delta L}{\delta \ell(z)} = \log \frac{\ell(z)}{q(z)} + 1 + \sum_{a=0}^{n} \beta_a z^a. \tag{A3}$$

Setting this to zero to find the minimum, the solution is

$$\ell(z) = q(z) \exp\left[-1 - \sum_{a=0}^{n} \beta_a z^a\right]. \tag{A4}$$

Writing $\ell(z) = g(z)q(z)$ and solving β_0 for the normalization condition, we recover the desired weight factor from Eq. (11), repeated for convenience

$$g(z) = \frac{1}{P}\exp\left[-\sum_{a=1}^{n}\beta_a z^a\right]. \tag{A5}$$

Here, the normalization factor is

$$P = \int q(z)\exp\left[-\sum_{a=1}^{n}\beta_a z^a\right]\mathrm{d}z. \tag{A6}$$

The remaining Lagrange multipliers β_a are determined by solving the moment constraints, which do not have a closed form in general.

APPENDIX B: ANALYTIC CLOSURE TESTS

In this appendix, we derive the asymptotic conditions under which moment unfolding will correctly recover the moments of the true distribution.

1. Perfect detector response

We start with the case of perfect detector response, such that we can work entirely with particle-level distributions in z. The derivation in Appendix A shows that the weight factor from Eq. (11) optimizes the relative entropy subject to the moment constraints. Here, we prove that minimizing the MLC loss with respect to the weight factor parameters recovers the desired moments. This is a nontrivial closure test of moment unfolding, since loss functions other than MLC do not generically satisfy this property.

Let $p(z)$ be the true particle-level distribution, $q(z)$ be the particle-level generator, and $g(z)$ be the weight factor from Eq. (11). For convenience, we define the reweighted distribution after moment unfolding as

$$\tilde{q}(x) = q(z)g(z). \tag{B1}$$

Asymptotically, the MLC loss of $p(z)$ to $\tilde{q}(z)$ is

$$L = -\int p(z)\log d(z)\mathrm{d}z - \int \tilde{q}(z)(1-d(z))\mathrm{d}z, \tag{B2}$$

where $d(z)$ is the discriminator function. Because we are assuming perfect detector response, it makes sense to talk about the discriminator acting on particle-level quantities.

Taking a functional derivative of the loss with respect to $d(z)$, we find

$$\frac{\delta L}{\delta d(z)} = -\frac{p(z)}{d(z)} + \tilde{q}(z). \tag{B3}$$

Setting this equal to zero, the optimal discriminator is $d_*(z) = p(z)/\tilde{q}(z)$. Plugging this back into the MLC loss and using the fact that $p(z)$ and $\tilde{q}(z)$ are both normalized, we find

$$L|_{d=d_*} = -\int p(z)\log\frac{p(z)}{\tilde{q}(z)}\mathrm{d}z = -D_{\mathrm{KL}}(p\|\tilde{q}). \tag{B4}$$

Note that this result is not symmetric between $p(z)$ and $\tilde{q}(z)$, which is essential to the rest of the derivation.

We now want to optimize the generator parameters assuming the optimal discriminator. Letting $\tilde{L} \equiv L|_{d=d_*}$ for notational convenience, the derivative of the discriminator-optimized loss with respect to the generator parameters is

$$\frac{\partial\tilde{L}}{\partial\beta_a} = \int \frac{\delta\tilde{L}}{\delta g(z)}\frac{\partial g(z)}{\partial\beta_a}\mathrm{d}z. \tag{B5}$$

The functional derivative of the loss with respect to the generator is

$$\frac{\delta\tilde{L}}{\delta g(z)} = \frac{p(z)}{g(z)}, \tag{B6}$$

while the derivatives of the generator with respect to its parameters are

$$\frac{\partial g(z)}{\partial\beta_a} = -g(z)\left(z^a + \frac{1}{P}\frac{\partial P}{\partial\beta_a}\right) = -g(z)(z^a - \langle Z^a\rangle_{\tilde{q}}). \tag{B7}$$

Therefore, setting Eq. (B5) equal to zero is equivalent to enforcing

$$\langle Z^a\rangle_{\tilde{q}} = \langle Z^a\rangle_p, \tag{B8}$$

for all $a = 1, \ldots, n$. This proves that the learned moments match those from data, at least in the asymptotic limit assuming perfect detector response and optimal learning.

2. Universal detector response

In the case of a realistic detector, the learned moments will not in general match the truth moments. That said, the deviations from closure will be small as long as detector distortions are small.

Crucially, all unfolding methods assume that the detector response is universal between real data and simulation. This means that the detector-level distributions can be written in terms of a universal response function $r(x|z)$ as

$$p(x) = \int r(x|z)p(z)\mathrm{d}z, \tag{B9}$$

$$q(x) = \int r(x|z)q(z)\mathrm{d}z. \tag{B10}$$

The reweighted distribution after moment unfolding is

$$\tilde{q}(x) = \int r(x|z)q(z)g(z)\mathrm{d}z. \tag{B11}$$

In the case that $r(x|z) = \delta(x - z)$, we recover the closure result from Eq. (B8). We now derive the moment relation in the case that the detector is imperfect but still universal.

The discriminator $d(x)$ now acts on detector-level quantities. Apart from swapping z for x, though, the derivation of the discriminator-optimized MLC loss is the same as in Eq. (B4)

$$\tilde{L} = -\int p(x) \log \frac{p(x)}{\tilde{q}(x)} \mathrm{d}x. \tag{B12}$$

The derivative of this with respect to the generator parameters is somewhat more involved than Eq. (B5)

$$\frac{\partial \tilde{L}}{\partial \beta_a} = \int \frac{\delta \tilde{L}}{\delta \tilde{q}(x)} \frac{\delta \tilde{q}(x)}{\delta g(z)} \frac{\partial g(z)}{\partial \beta_a} \mathrm{d}z \mathrm{d}x. \tag{B13}$$

The functional derivatives are

$$\frac{\delta \tilde{L}}{\delta \tilde{q}(x)} = \frac{p(x)}{\tilde{q}(x)}, \tag{B14}$$

$$\frac{\delta \tilde{q}(x)}{\delta g(z)} = r(x|z) q(z). \tag{B15}$$

The $\partial g(z)/\partial \beta_a$ derivative is the same as Eq. (B7).

Setting Eq. (B13) equal to zero, we find that the learned moments satisfy

$$\langle Z^a \rangle_{\tilde{q}} = \langle Z^a \rangle_{p_{\mathrm{mod}}}, \tag{B16}$$

where the modified data distribution is

$$p_{\mathrm{mod}}(z) = \int \frac{p(x) r(x|z) \tilde{q}(z)}{\tilde{q}(x)} \mathrm{d}x. \tag{B17}$$

Thus, the moments of $\tilde{q}(z)$ match the moments of $p_{\mathrm{mod}}(z)$, which are in general different from those of $p(z)$.

To better interpret $p_{\mathrm{mod}}(z)$, it is convenient to rewrite it in the following form:

$$p_{\mathrm{mod}}(z) = \int f(z|z') p(z') \mathrm{d}z', \tag{B18}$$

where $f(z|z')$ can be thought of as a transfer function that maps the actual particle-level truth information to the learned particle-level information. In the case of perfect detector response, $f(z|z') = \delta(z - z')$. For a realistic detector, one can manipulate Eq. (B17) to find

$$f(z|z') = \int \tilde{q}(z|x) r(x|z') \mathrm{d}x. \tag{B19}$$

Here, $\tilde{q}(z|x) = r(x|z)\tilde{q}(z)/\tilde{q}(x)$ is the *inverse* detector response derived from the reweighted simulation, which is in general not universal.

Thus, one can interpret $f(z|z')$ as taking the particle-level information, passing it through the universal detector response, and pulling it back through the nonuniversal reweighted simulation. To the extent that the reweighted simulation is sufficiently similar to the real data, $p_{\mathrm{mod}}(z)$ will be similar enough to $p(z)$ to satisfy closure. One obstruction to closure is if the detector response includes large distortions, such that the inverse detector response is highly dependent on the reweighting function. Another obstruction is if the particle-level generator has poor overlapping support with the truth, such that the reweighting function needs to be large in poorly modeled regions of phase space. Both of these obstructions are common to all unfolding methods, though, and not unique to moment unfolding.

[1] G. Altarelli and G. Parisi, Nucl. Phys. **B126**, 298 (1977).
[2] Y. L. Dokshitzer, Sov. Phys. JETP **46**, 641 (1977), https://inspirehep.net/literature/126153.
[3] V. N. Gribov and L. N. Lipatov, Sov. J. Nucl. Phys. **15**, 438 (1972), https://inspirehep.net/literature/73449.
[4] Y. K. Li *et al.* (AMY Collaboration), Phys. Rev. D **41**, 2675 (1990).
[5] W. Braunschweig *et al.* (TASSO Collaboration), Z. Phys. C **47**, 187 (1990).
[6] P. A. Movilla Fernandez, O. Biebel, S. Bethke, S. Kluth, and P. Pfeifenschneider (JADE Collaboration), Eur. Phys. J. C **1**, 461 (1998).
[7] K. Ackerstaff *et al.* (OPAL Collaboration), Z. Phys. C **75**, 193 (1997).
[8] P. Abreu *et al.* (DELPHI Collaboration), Phys. Lett. B **456**, 322 (1999).
[9] M. Acciarri *et al.* (L3 Collaboration), Phys. Lett. B **489**, 65 (2000).
[10] DELPHI Collaboration, Eur. Phys. J. C **29**, 285 (2003).
[11] DELPHI Collaboration, Eur. Phys. J. C **37**, 1 (2004).
[12] P. Achard *et al.* (L3 Collaboration), Phys. Rep. **399**, 71 (2004).
[13] ALEPH Collaboration, Eur. Phys. J. C **35**, 457 (2004).
[14] OPAL Collaboration, Eur. Phys. J. C **40**, 287 (2005).
[15] C. Pahl, S. Bethke, S. Kluth, and J. Schieck (JADE Collaboration), Eur. Phys. J. C **60**, 181 (2009); **62**, 451(E) (2009).
[16] R. Abbate, M. Fickinger, A. H. Hoang, V. Mateu, and I. W. Stewart, Phys. Rev. D **86**, 094002 (2012).

[17] G. Zech and B. Aslan, PHYSTAT (2003), https://inspirehep.net/literature/637561.
[18] L. Lindemann and G. Zech, Nucl. Instrum. Methods Phys. Res., Sect. A **354**, 516 (1995).
[19] K. Datta, D. Kar, and D. Roy, arXiv:1806.00433.
[20] M. Bunse, N. Piatkowski, T. Ruhe, W. Rhode, and K. Morik, in *Proceedings of the 5th International Conference on Data Science and Advanced Analytics (DSAA)* (IEEE, New York, 2018), pp. 21–30.
[21] T. Ruhe, T. Voigt, M. Wornowizki, M. Börner, W. Rhode, and K. Morik (2019), https://ui.adsabs.harvard.edu/abs/2019ASPC..521..394R/abstract.
[22] A. Andreassen, P. T. Komiske, E. M. Metodiev, B. Nachman, and J. Thaler, Phys. Rev. Lett. **124**, 182001 (2020).
[23] M. Bellagente, A. Butter, G. Kasieczka, T. Plehn, and R. Winterhalder, SciPost Phys. **8**, 070 (2020).
[24] M. Bellagente, A. Butter, G. Kasieczka, T. Plehn, A. Rousselot, and R. Winterhalder, SciPost Phys. **9**, 074 (2020).
[25] M. Vandegar, M. Kagan, A. Wehenkel, and G. Louppe, in *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, Proceedings of Machine Learning Research Vol. 130, edited by A. Banerjee and K. Fukumizu (PMLR, Cambridge, MA, 2021), pp. 2107–2115, arXiv:2011.05836.
[26] A. Andreassen, P. T. Komiske, E. M. Metodiev, B. Nachman, A. Suresh, and J. Thaler, arXiv:2105.04448.
[27] J. N. Howard, S. Mandt, D. Whiteson, and Y. Yang, Sci. Rep. **12**, 7567 (2022).
[28] M. Backes, A. Butter, M. Dunford, and B. Malaescu, SciPost Phys. Core **7**, 007 (2024).
[29] M. Arratia, D. Britzger, O. Long, and B. Nachman, J. Instrum. **17**, P07009 (2022).
[30] J. Chan and B. Nachman, Phys. Rev. D **108**, 016002 (2023).
[31] A. Shmakov, K. Greif, M. Fenton, A. Ghosh, P. Baldi, and D. Whiteson, arXiv:2305.10399.
[32] T. Alghamdi *et al.*, Phys. Rev. D **108**, 094030 (2023).
[33] S. Diefenbacher, G.-H. Liu, V. Mikuni, B. Nachman, and W. Nie, Phys. Rev. D **109**, 076011 (2024).
[34] C.-C. Pan, X. Dong, Y.-C. Sun, A.-Y. Cheng, A.-B. Wang, Y.-X. Hu, and H. Cai, arXiv:2406.01635.
[35] M. Arratia *et al.*, J. Instrum. **17**, P01024 (2022).
[36] N. Huetsch *et al.*, arXiv:2404.18807.
[37] V. Andreev *et al.* (H1 Collaboration), Phys. Rev. Lett. **128**, 132002 (2022).
[38] B. Nachman, https://inspirehep.net/literature/2136855.
[39] V. Andreev *et al.* (H1 Collaboration), Phys. Lett. B **844**, 138101 (2023).
[40] V. Andreev *et al.* (H1 Collaboration), https://inspirehep.net/literature/1912842.
[41] LHCb Collaboration, Phys. Rev. D **108**, L031103 (2023).
[42] P. T. Komiske, S. Kryhin, and J. Thaler, Phys. Rev. D **106**, 094021 (2022).
[43] CMS Collaboration, Measurement of event shapes in minimum bias events from pp collisions at13 TeV, Technical Report No. CMS-PAS-SMP-23-008, CERN, Geneva, 2024, https://cds.cern.ch/record/2899591.
[44] Y. Song (STAR Collaboration), arXiv:2307.07718.
[45] G. Aad *et al.* (ATLAS Collaboration), arXiv:2405.20041.
[46] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, arXiv:1406.2661.
[47] J. Chan, X. Ju, A. Kania, B. Nachman, V. Sangli, and A. Siodmok, J. High Energy Phys. 09 (2023) 084.
[48] C. Bierlich, P. Ilten, T. Menzo, S. Mrenna, M. Szewc, M. K. Wilkinson, A. Youssef, and J. Zupan, SciPost Phys. **17**, 045 (2024).
[49] A. Buckley, L. Corpe, M. Filipovich, C. Gutschow, N. Rozinsky, S. Thor, Y. Yeh, and J. Yellen, arXiv:2312.15070.
[50] G. D'Agostini, Nucl. Instrum. Methods Phys. Res., Sect. A **362**, 487 (1995).
[51] W. H. Richardson, J. Opt. Soc. Am. **62**, 55 (1972).
[52] L. B. Lucy, Astron. J. **79**, 745 (1974).
[53] A. Hocker and V. Kartvelishvili, Nucl. Instrum. Methods Phys. Res., Sect. A **372**, 469 (1996).
[54] S. Schmitt, J. Instrum. **7**, T10003 (2012).
[55] G. Cowan, Conf. Proc. C **0203181**, 248 (2002), https://inspirehep.net/literature/599644.
[56] V. Blobel, in PHYSTAT2011 Proceedings (2011), p. 240, 10.5170/CERN-2011-006.
[57] V. Blobel, *Data Analysis in High Energy Physics* (2013), p. 187, 10.1002/9783527653416.ch6.
[58] R. Balasubramanian, L. Brenner, C. Burgard, G. Cowan, V. Croft, W. Verkerke, and P. Verschuuren, Int. J. Mod. Phys. A **35**, 2050145 (2020).
[59] K. Sharp and F. Matschinsky, Entropy **17**, 1971 (2015).
[60] R. T. D'Agnolo and A. Wulzer, Phys. Rev. D **99**, 015014 (2019).
[61] R. T. D'Agnolo, G. Grosso, M. Pierini, A. Wulzer, and M. Zanetti, Eur. Phys. J. C **81**, 89 (2021).
[62] B. Nachman and J. Thaler, Phys. Rev. D **103**, 116013 (2021).
[63] Z. I. Botev and D. P. Kroese, Methodol. Comput. Appl. Probab. **13**, 1 (2011).
[64] M. Erdmann, B. Fischer, D. Noll, Y. Alexander Rath, M. Rieger, and D. Josef Schmidt, J. Phys. Conf. Ser. **1525**, 012094 (2020).
[65] F. Chollet, Keras, https://github.com/fchollet/keras (2017).
[66] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, in *OSDI* (USENIX Association, Savannah, GA, 2016), Vol. 16, pp. 265–283.
[67] D. Kingma and J. Ba, arXiv:1412.6980.
[68] A. J. Larkoski, I. Moult, and B. Nachman, Phys. Rep. **841**, 1 (2020).
[69] R. Kogler *et al.*, Rev. Mod. Phys. **91**, 045003 (2019).
[70] D. Krohn, M. D. Schwartz, T. Lin, and W. J. Waalewijn, Phys. Rev. Lett. **110**, 212001 (2013).
[71] A. J. Larkoski, S. Marzani, and J. Thaler, Phys. Rev. D **91**, 111501 (2015).
[72] M. Dasgupta, A. Fregoso, S. Marzani, and G. P. Salam, J. High Energy Phys. 09 (2013) 029.
[73] A. J. Larkoski, S. Marzani, G. Soyez, and J. Thaler, J. High Energy Phys. 05 (2014) 146.
[74] S. Acharya *et al.* (A Large Ion Collider Experiment and ALICE Collaborations), Phys. Rev. Lett. **128**, 102001 (2022).
[75] W. J. Waalewijn, Phys. Rev. D **86**, 094030 (2012).
[76] H. T. Li and I. Vitev, Phys. Rev. D **101**, 076020 (2020).

[77] Z.-B. Kang, X. Liu, S. Mantry, M. C. Spraker, and T. Wilson, Phys. Rev. D **103**, 074028 (2021).
[78] Z.-B. Kang, X. Liu, S. Mantry, and D. Y. Shao, Phys. Rev. Lett. **125**, 242003 (2020).
[79] G. Aad *et al.* (ATLAS Collaboration), Phys. Rev. D **93**, 052003 (2016).
[80] A. M. Sirunyan *et al.* (CMS Collaboration), J. High Energy Phys. 10 (2017) 131.
[81] A. M. Sirunyan *et al.* (CMS Collaboration), J. High Energy Phys. 07 (2020) 115.
[82] A. Andreassen, P. Komiske, E. Metodiev, B. Nachman, and J. Thaler, Pythia/Herwig + DELPHES jet datasets for OmniFold unfolding (2019), 10.5281/zenodo.3548091.
[83] M. Bahr *et al.*, Eur. Phys. J. C **58**, 639 (2008).
[84] J. Bellm *et al.*, Eur. Phys. J. C **76**, 196 (2016).
[85] J. Bellm *et al.*, arXiv:1705.06919.
[86] The ATLAS Collaboration, ATLAS Run 1 Pythia8 tunes, Technical Report No. ATL-PHYS-PUB-2014-021, CERN, Geneva, 2014, https://cds.cern.ch/record/1966419.
[87] T. Sjöstrand, S. Mrenna, and P. Z. Skands, Comput. Phys. Commun. **178**, 852 (2008).
[88] T. Sjöstrand, S. Mrenna, and P. Z. Skands, J. High Energy Phys. 05 (2006) 026.
[89] T. Sjöstrand, S. Ask, J. R. Christiansen, R. Corke, N. Desai, P. Ilten, S. Mrenna, S. Prestel, C. O. Rasmussen, and P. Z. Skands, Comput. Phys. Commun. **191**, 159 (2015).
[90] J. de Favereau, C. Delaere, P. Demin, A. Giammanco, V. Lemaître, A. Mertens, and M. Selvaggi (DELPHES 3 Collaboration), J. High Energy Phys. 02 (2014) 057.
[91] A. M. Sirunyan *et al.* (CMS Collaboration), J. Instrum. **12**, P10003 (2017).
[92] M. Cacciari, G. P. Salam, and G. Soyez, J. High Energy Phys. 04 (2008) 063.
[93] M. Cacciari, G. P. Salam, and G. Soyez, Eur. Phys. J. C **72**, 1896 (2012).
[94] M. Cacciari and G. P. Salam, Phys. Lett. B **641**, 57 (2006).
[95] O. Kitouni, N. Nolte, and M. Williams, Mach. Learn. Sci. Tech. **4**, 035020 (2023).
[96] S. Bright-Thonney, B. Nachman, and J. Thaler, Phys. Rev. D **110**, 014029 (2024).
[97] ATLAS Collaboration, Phys. Rev. D **93**, 052003 (2016).
[98] http://iaifi.org/.
[99] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, and C. Willing, in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, edited by F. Loizides and B. Schmidt (IOS Press, Amsterdam, The Netherlands, 2016), pp. 87–90.
[100] https://github.com/HEP-GAN/MomentUnfolding.
[101] C. R. Harris *et al.*, Nature (London) **585**, 357 (2020).
[102] J. D. Hunter, Comput. Sci. Eng. **9**, 90 (2007).