Title
OmniFold: A Method to Simultaneously Unfold All Observables.

Permalink
https://escholarship.org/uc/item/3vp4m5wf

Journal
Physical review letters, 124(18)

ISSN
0031-9007

Authors
Andreassen, Anders
Komiske, Patrick T
Metodiev, Eric M
et al.

Publication Date
2020-05-01

DOI
10.1103/physrevlett.124.182001

Peer reviewed

OmniFold: A Method to Simultaneously Unfold All Observables

Anders Andreassen,[1,2,3,*] Patrick T. Komiske,[4,†] Eric M. Metodiev,[4,‡] Benjamin Nachman,[2,§] and Jesse Thaler[4,∥]
[1]*Department of Physics, University of California, Berkeley, California 94720, USA*
[2]*Physics Division, Lawrence Berkeley National Laboratory, Berkeley, California 94720, USA*
[3]*Google, Mountain View, California 94043, USA*
[4]*Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA*

(Received 2 December 2019; accepted 3 April 2020; published 7 May 2020)

Collider data must be corrected for detector effects ("unfolded") to be compared with many theoretical calculations and measurements from other experiments. Unfolding is traditionally done for individual, binned observables without including all information relevant for characterizing the detector response. We introduce OmniFold, an unfolding method that iteratively reweights a simulated dataset, using machine learning to capitalize on all available information. Our approach is unbinned, works for arbitrarily high-dimensional data, and naturally incorporates information from the full phase space. We illustrate this technique on a realistic jet substructure example from the Large Hadron Collider and compare it to standard binned unfolding methods. This new paradigm enables the simultaneous measurement of all observables, including those not yet invented at the time of the analysis.

DOI: 10.1103/PhysRevLett.124.182001

Measuring properties of particle collisions is a central goal of particle physics experiments, such as those at the Large Hadron Collider (LHC). After correcting for detector effects, distributions of collider observables at "truth level" can be compared with semi-inclusive theoretical predictions as well as with measurements from other experiments. These comparisons are widely used to enhance our understanding of the Standard Model, tune parameters of Monte Carlo event generators, and enable precision searches for new physics. "Unfolding" is the process of obtaining these truth distributions (particle-level) from measured information recorded by a detector (detector-level). The unfolding process ensures that measurements are independent of the specific experimental context, allowing for comparisons across different experiments and usage with the latest theoretical tools [1], even long after the original analysis is completed. Many unfolding methods have been proposed and are currently used by experiments. See Refs. [2–5] for reviews and Refs. [6–8] for the most widely used unfolding algorithms.

Current unfolding methods face three key challenges. First, all of the widely used methods require the measured observables to be binned into histograms. This binning must be determined ahead of time and is often chosen manually. Second, because the measurements are binned, one can only unfold a small number of observables simultaneously. Multidifferential cross section measurements beyond two or three dimensions are simply not feasible. Finally, unfolding corrections for detector effects often do not take into account all possible auxiliary features that control the detector response. Even though the inputs to the unfolding can be calibrated, if the detector response depends on features that are not used directly in the unfolding, then the results will be suboptimal and potentially biased.

This Letter introduces OmniFold, a new approach that solves all three of these unfolding challenges. Detector-level quantities are iteratively unfolded, using machine learning to handle phase space of any dimensionality without requiring binning. Utilizing the full phase space information mitigates the problem of auxiliary features controlling the detector response. There have been previous proposals to use machine learning methods for unfolding [9–11] as well as proposals to perform unfolding without binning [10–13]. These proposals, however, are untenable in high dimensions and do not reduce to standard methods in the binned case. OmniFold naturally processes high-dimensional features, in the spirit of previous machine-learning-based reweighting strategies [14–19], and it reduces to well-established methods [6] in the binned case. We also introduce simpler versions of the procedure, using single or multiple observables, named UniFold and MultiFold, respectively.

All unfolding methods require a trustable detector simulation to estimate the detector response. In the binned formulation, the folding equation can be written as $\mathbf{m} = \mathbf{R}\mathbf{t}$, where $\mathbf{m}$ and $\mathbf{t}$ are vectors of the measured detector-level and true particle-level histograms, respectively. **R** is the "response matrix"

$$R_{ij} = \Pr(\text{measure } i|\text{truth is } j). \tag{1}$$

In general, **R** is not invertible, so the unfolding problem has no unique solution, and methods attempt to achieve a useful solution in various ways. One of the most widely used methods is iterative Bayesian unfolding (IBU) [6], also known as Richardson-Lucy deconvolution [20,21]. Given a measured spectrum $m_i = \Pr(\text{measure } i)$ and a prior spectrum $t_j^{(0)} = \Pr_0(\text{truth is } j)$, IBU proceeds iteratively according to the equation:

$$\begin{aligned} t_j^{(n)} &= \sum_i \Pr_{n-1}(\text{truth is } j|\text{measure } i)\Pr(\text{measure } i) \\ &= \sum_i \frac{R_{ij}t_j^{(n-1)}}{\sum_k R_{ik}t_k^{(n-1)}} \times m_i, \end{aligned} \tag{2}$$

where n is the iteration number.

OmniFold uses machine learning to generalize Eq. (2) to the unbinned, full phase space. A key concept for this approach is the likelihood ratio

$$L[(w,X),(w',X')](x) = \frac{p_{(w,X)}(x)}{p_{(w',X')}(x)}, \tag{3}$$

where $p_{(w,X)}$ is the probability density of x estimated from empirical weights w and samples X. The function $L[(w,X),(w',X')](x)$ can be approximated using a classifier trained to distinguish (w,X) from (w',X'). This property has been successfully exploited using neural networks for full phase-space Monte Carlo reweighting and parameter estimation [19,22–26]. Here, we use neural network classifiers to iteratively reweight the particle- and detector-level Monte Carlo weights, resulting in an unfolding procedure.

The OmniFold technique is illustrated in Fig. 1. Intuitively, synthetic detector-level events ("simulation") are reweighted to match experimental data ("data"), and then the reweighted synthetic events, now evaluated at particle-level ("generation"), are further reweighted to estimate the true particle-level information ("truth"). The starting point is a synthetic Monte Carlo dataset composed of pairs (t,m), where each particle-level event t is pushed through the detector simulation to obtain a detector-level event m. Particle-level events have initial weights $\nu_0(t)$, and when t is pushed to m, these become detector-level weights $\nu_0^{\text{push}}(m) = \nu_0(t)$. OmniFold iterates the following steps: 1. $\omega_n(m) = \nu_{n-1}^{\text{push}}(m)L[(1,\text{Data}),(\nu_{n-1}^{\text{push}},\text{Sim.})](m)$, 2. $\nu_n(t) = \nu_{n-1}(t)L[(\omega_n^{\text{pull}},\text{Gen.}),(\nu_{n-1},\text{Gen.})](t)$. The first step yields new detector-level weights $\omega_n(m)$, which are pulled back to particle-level weights $\omega_n^{\text{pull}}(t) = \omega_n(m)$ using the same synthetic pairs (t,m). Note that ν^{push} and ω^{pull} are not, strictly speaking, functions because of the multivalued nature of the detector simulation. The second step ensures that ν_n is a valid weighting function of the particle-level quantities.

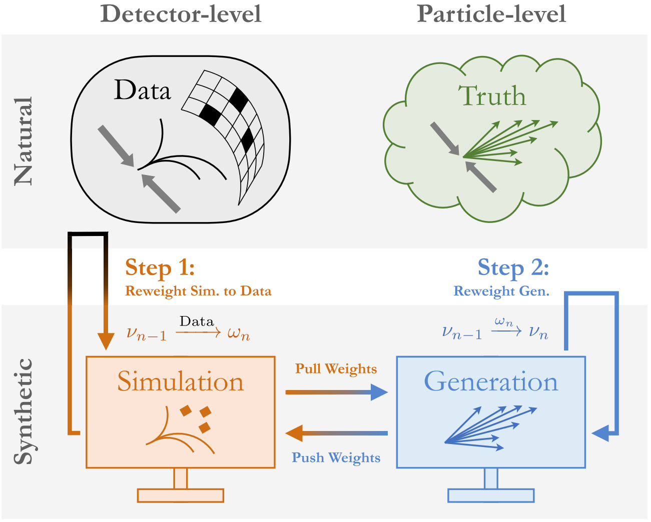

FIG. 1. An illustration of OmniFold, applied to a set of synthetic and natural data. As a first step, starting from prior weights ν_0, the detector-level synthetic data ("simulation") is reweighted to match the detector-level natural data (simply "data"). These weights ω_1 are pulled back to induce weights on the particle-level synthetic data ("generation"). As a second step, the initial generation is reweighted to match the new weighted generation. The resulting weights ν_1 are pushed forward to induce a new simulation, and the process is iterated.

Assuming $\nu_0(t) = 1$, in the first iteration step 1 learns $\omega_1(m) = p_{\text{Data}}(m)/p_{\text{Sim}}(m)$, which is pulled back to the particle-level weights $\omega_1^{\text{pull}}(t)$. Step 2 simply converts the per-instance weights $\omega_1^{\text{pull}}(t)$ to a valid particle-level weighting function $\nu_1(t)$. After one iteration, the new induced truth is

$$\nu_1(t)p_{\text{Gen}}(t) = \int dm' p_{\text{Gen|Sim}}(t|m')p_{\text{Data}}(m'). \tag{4}$$

This is a continuous version of IBU from Eq. (2), where the sum has been promoted to a full phase-space integral. In fact, OmniFold (and IBU) are iterative strategies that converge to the maximum likelihood estimate of the true particle-level distribution [27–31], which we discuss in detail in the Supplemental Material [32]. After n iterations, the unfolded distribution is

$$p_{\text{unfolded}}^{(n)}(t) = \nu_n(t)p_{\text{Gen}}(t). \tag{5}$$

The unfolded result can be presented either as a set of generated events $\{t\}$ with weights $\{\nu_n(t)\}$ (and uncertainties) or, more compactly, as the learned weighting function ν_n and instructions for sampling from p_{Gen}.

To demonstrate the versatility and power of OmniFold, we perform a proof-of-concept study relevant for the LHC. Specifically, we unfold the full radiation pattern (i.e., full phase space) of jets, which are collimated sprays of particles arising from the fragmentation and hadronization of high-energy quarks and gluons. Jets are an ideal environment in which to benchmark unfolding techniques, since detector effects often account for a significant portion of the experimental measurement uncertainties for many jet substructure observables [33]. With the radiation pattern unfolded, one can obtain the unfolded distribution of *any* observable using Eq. (5). Hence, this procedure can be viewed as simultaneously unfolding all observables.

Our study is based on proton-proton collisions generated at $\sqrt{s} = 14$ TeV with the default tune of HERWIG 7.1.5 [34–36] and Tune 26 [37] of PYTHIA 8.243 [38–40] in order to study a challenging setting where the "natural" and "synthetic" distributions are substantially different. As a proxy for detector effects and a full detector simulation, we use the DELPHES 3.4.2 [41] fast simulation of the CMS detector, which uses particle flow reconstruction. 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 [42] implemented in FastJet 3.3.2 [43,44]. One of the simulations (HERWIG) plays the role of data and truth, while the other (PYTHIA) is used to derive the unfolding corrections. To reduce acceptance effects, the leading jets are studied in events with a Z boson with transverse momentum $p_T^Z > 200$ GeV. After applying the selections, we obtain approximately 1.6×10^6 events from each generator.

Any suitable machine learning architecture can be used for OmniFold. For this study, we use particle flow networks (PFNs) [45,46] to process jets in their natural representation as sets of particles. Intuitively, PFNs learn and processes a set of additive observables via $\text{PFN}(\{p_i\}_{i=1}^M) = F(\sum_{i=1}^M \Phi(p_i))$ for an event with M particles p_i, where F and Φ are parametrized by fully connected networks. We specify the particles by their transverse momentum p_T, rapidity y, azimuthal angle ϕ, and particle identification code [47], restricted to the experimentally accessible information (PFN-Ex [45]) at detector level. To define separate models for step 1 and step 2, we use the PFN architecture and training parameters of Ref. [45] with latent space dimension $\ell = 256$, implemented in the ENERGYFLOW PYTHON package [48]. Neural networks are trained with Keras [49] and TensorFlow [50] using the Adam [51] optimization algorithm. The models are randomly initialized in the first iteration and subsequently warm started using the model from the previous iteration. 20% of the events are reserved as a validation set during training.

To investigate the unfolding performance, we consider six widely used jet substructure observables [52]. The first four are jet mass m, constituent multiplicity M, the N-subjettiness ratio $\tau_{21} = \tau_2^{(\beta=1)}/\tau_1^{(\beta=1)}$ [53,54], and the jet width w (implemented as $\tau_1^{(\beta=1)}$). Since jet grooming [55–59] is of recent interest, we also show the jet mass $\ln\rho = \ln m_{\text{SD}}^2/p_T^2$ and momentum fraction z_g after Soft Drop grooming [58,59] with $z_{\text{cut}} = 0.1$ and $\beta = 0$. Several of these observables are computed with the help of FastJet Contrib 1.042 [60].

The unfolding performance of OmniFold is shown in Fig. 2 and compared to IBU, both with $n = 5$ iterations. We found little difference between $n = 3$ and $n = 5$, though OmniFold exhibits a slight preference for more iterations. OmniFold succeeds in simultaneously unfolding all of these observables, achieving performance comparable to or better than IBU applied to each observable individually. The mass is challenging for all methods as particle-type information is relevant at the particle level but is not fully known at the detector level, introducing additional prior dependence. Though OmniFold is unbinned, the data are only able to constrain energy and angular scales comparable to the detector resolution.

Statistical uncertainties from the prior distribution are shown in the bottom panels of Fig. 2, holding the unfolding procedure (i.e., response matrix and reweighting) fixed. For this proof-of-concept study, we do not show systematic uncertainties, though the procedure for deriving them is the same as for IBU. Nonclosure and modeling uncertainties can be derived in the standard way by testing the procedure on different Monte Carlo samples and comparing the results to the known truth distributions. (We checked that OmniFold satisfies technical closure when PYTHIA is unfolded to itself.) Experimental systematic uncertainties can be obtained by varying the relevant effects and repeating the unfolding procedure. Like other unfolding procedures, OmniFold cannot improve the results in phase-space regions that are unconstrained by observed quantities. It can, however, improve the performance if the full phase space contains auxiliary features relevant for the detector response. To capitalize on this full phase-space approach, it is essential that the detector simulation properly describes these features and that systematic uncertainties are estimated using a high-dimensional approach [61,62].

To highlight the flexibility of our unfolding framework, we study variations of OmniFold, where the available information is varied by controlling the inputs: (i) UniFold: A single observable as input. This is an unbinned version of IBU. (ii) MultiFold: Many observables as input. Here, we use the six jet substructure observables in Fig. 2 to derive the detector response. (iii) OmniFold: The full event (or jet) as input, using the full phase space information. The unfolding performance of each method on our six substructure observables is tabulated in Table. I and compared to IBU. The UniFold and MultiFold implementations both use dense

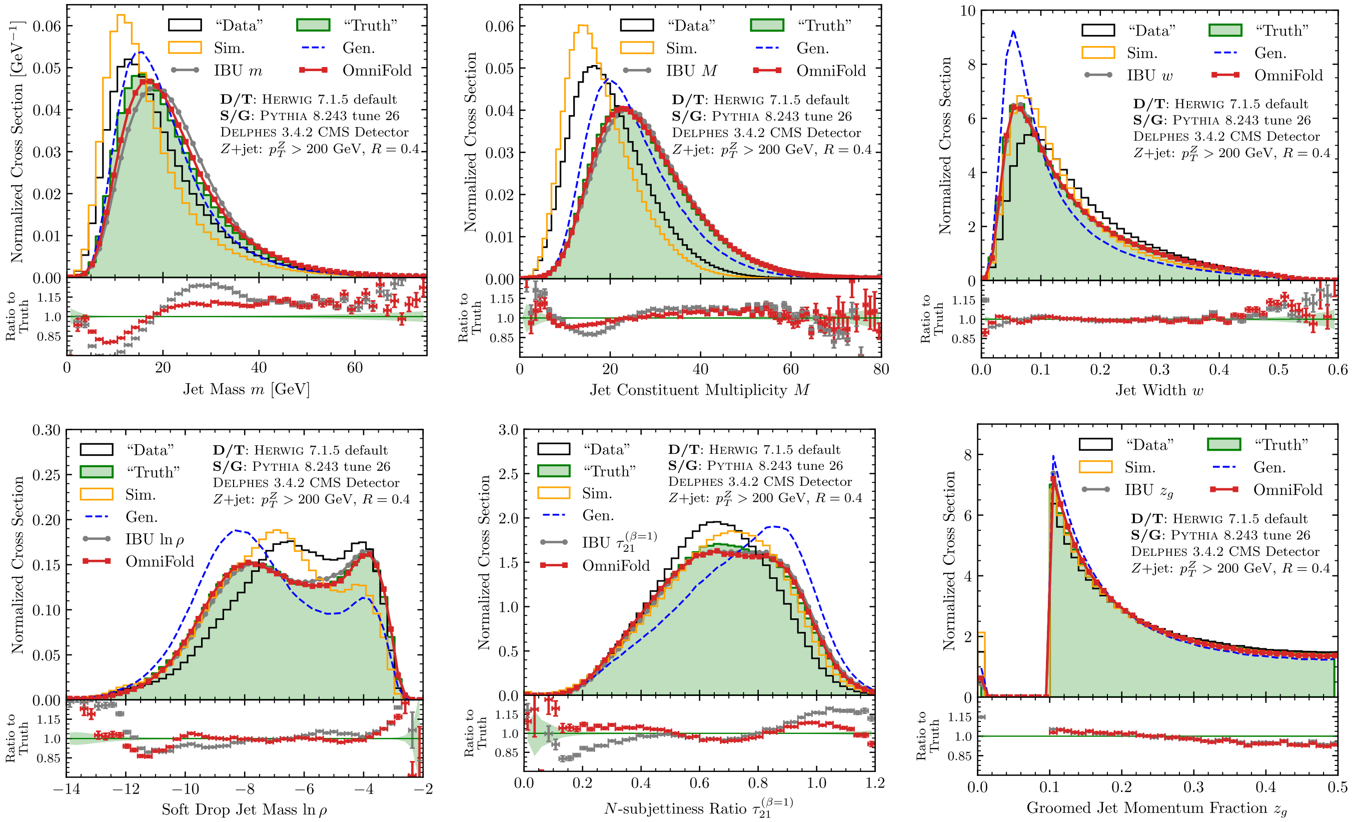

FIG. 2. The unfolding results for six jet substructure observables, using HERWIG 7.1.5 ("Data"/"Truth") and PYTHIA 8.243 tune 26 (Sim./Gen.), unfolded with OMNIFOLD and compared to IBU. OMNIFOLD matches or exceeds the unfolding performance of IBU on all of these observables. We emphasize that OMNIFOLD is a single general unfolding procedure, whereas unfolding with IBU must be done observable by observable. Statistical uncertainties are shown only in the ratio panel.

networks with three layers of one hundred nodes each and a two-node output layer. We see good unfolding performance across all methods, and even though OMNIFOLD is not directly trained on these six observables, it performs comparably to or better than MULTIFOLD. While the detector response depends on the jet rapidity, we checked that MULTIFOLD did not significantly benefit from including the rapidity, though doing so could be important in a real experimental context. In general, additional information can be included and the unfolding procedure can be repeated, with the final model chosen as the one with the best detector-level agreement with the data.

TABLE I. The unfolding performance of OMNIFOLD, MULTIFOLD, and UNIFOLD on six jet substructure observables, compared to IBU. The performance is quantified by the triangular discriminator [63–65] $\Delta(p,q) = \frac{1}{2}\int d\lambda\{[p(\lambda) - q(\lambda)]^2/p(\lambda) + q(\lambda)\}$ $(\times 10^3)$ between the unfolded and truth-level (binned) histograms. Also shown are the distances from data (no unfolding) and generation (the prior). The best unfolding method for each observable is shown in bold. All methods perform well, with OMNIFOLD providing consistently good performance.

	Observable					
Method	m	M	w	$\ln\rho$	τ_{21}	z_g
OMNIFOLD	**2.77**	**0.33**	0.10	**0.35**	0.53	0.68
MULTIFOLD	3.80	0.89	**0.09**	0.37	**0.26**	**0.15**
UNIFOLD	8.82	1.46	0.15	0.59	1.11	0.59
IBU	9.31	1.51	0.11	0.71	1.10	0.37
Data	24.6	130	15.7	14.2	11.1	3.76
Generation	3.62	15	22.4	19	20.8	3.84

Since OMNIFOLD unfolds the full radiation pattern, it can be used to probe new, physically interesting quantities that are challenging to unfold with existing methods. One example is the recently proposed correlation (fractal) dimension of jets [66,67], which is a function of the energy scale Q. This complicated statistic is defined by pairwise metric distances between jet radiation patterns, falling outside of the purview of single-observable unfolding techniques. Within our jet samples, we restrict to energetic jets with $p_T^{\mathrm{jet}} > 500$ GeV, boosted to the origin of the rapidity-azimuth plane, and with constituents rescaled to have p_T summing to 500 GeV. The correlation dimensions of these jets, both before and after applying OMNIFOLD, are shown in Fig. 3. The unfolded results match the true distribution over a wide range of Q values, with residual prior dependence seen at low Q (i.e., the infrared) where

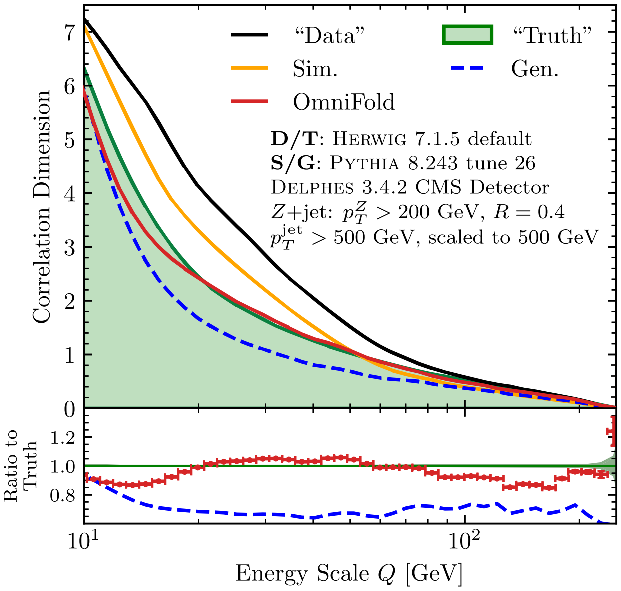

FIG. 3. The correlation dimension of the space of jets, unfolded with OmniFold. The unfolded results closely match the truth-level dimension over most of the energy range, tending toward the prior in the more difficult phase space region at low Q. Unfolding a complicated statistic such as the correlation dimension is challenging with standard methods.

jets have a higher dimensionality and detector effects have a larger impact, thus making the unfolding problem more difficult. More broadly, OmniFold opens the door to going beyond per-event collider observables towards more nuanced or intricate measurements of the data.

In conclusion, we have presented a potentially transformative unfolding paradigm based on iteratively reweighting a set of simulated events with machine learning. Our OmniFold approach allows an entire dataset to be unfolded using all of the available information, avoiding the need for binning and restricting to single observables. We have demonstrated the power of this method in a (simulated) case of interest by unfolding the full radiation pattern of jets, paving the way for significant advances in jet substructure at the LHC. Our unfolding framework allowed us to go beyond per-event observables towards unfolding more complex dataset statistics, such as fractal dimensions of the space of jets. Going even further, (unsupervised) machine learning models may be trained directly at the particle level by using the unfolded and weighted dataset, which is a fascinating avenue for further exploration. These advances have broad applicability beyond particle physics in domains where deconvolution or unfolding is used, such as image-based measurements and quantum computation [68]. To enable future unfolding studies and developments, we have made our code and jet datasets publicly available [69,70], including two additional tunes of pythia beyond those presented here. Finally, our reweighting-based unfolding strategy allows for new observables to be measured long after the unfolding is carried out, which can significantly empower future public and archival collider data analyses [71].

We are grateful to Gabriel Collin, Christian Herwig, Andrew Larkoski, Matthew LeBlanc, Salvatore Rappoccio, Jennifer Roloff, Matthew Schwartz, Daniel Whiteson, and Mike Williams for helpful comments and discussions. We also thank the referees for their valuable suggestions, including the clarification of the statistical framing of the method. The authors benefited from the hospitality of the Harvard Center for the Fundamental Laws of Nature. This work was partially completed at the Aspen Center for Physics, which is supported by National Science Foundation Grant No. PHY-1607611. A. A. and B. N. were supported by the U.S. Department of Energy (DOE), Office of Science under Contract No. DE-AC02-05CH11231. P. T. K., E. M. M., and J. T. were supported by the DOE Office of Nuclear Physics under Grant No. DE-SC0011090 and the DOE Office of High Energy Physics under Grants No. DE-SC0012567 and No. DE-SC0019128. J. T. is additionally supported by the Simons Foundation through a Simons Fellowship in Theoretical Physics. Cloud computing resources were provided through a Google Cloud allotment from the MIT Quest for Intelligence. B. N. would like to thank NVIDIA for providing Volta GPUs for neural network training.

[*]ajandreassen@google.com
[†]pkomiske@mit.edu
[‡]metodiev@mit.edu
[§]bpnachman@lbl.gov
[‖]jthaler@mit.edu

[1] For fully exclusive theoretical predictions, one could alternatively forward fold to compare to experimental data.

[2] G. Cowan, A survey of unfolding methods for particle physics, Conf. Proc. **C0203181**, 248 (2002).

[3] V. Blobel, Unfolding methods in particle physics, in *PHYSTAT 2011, Geneva, Switzerland, 2011* (CERN, Geneva, 2011), p. 240–251, http://dx.doi.org/10.5170/CERN-2011-006.

[4] V. Blobel, Unfolding, in *Data Analysis in High Energy Physics: A Practical Guide to Statistical Methods* (John Wiley & Sons, Ltd., Hoboken, 2013), pp. 187–225, http://dx.doi.org/10.1002/9783527653416.ch6.

[5] R. Balasubramanian, L. Brenner, C. Burgard, G. Cowan, V. Croft, W. Verkerke, and P. Verschuuren, Statistical method and comparison of different unfolding techniques using RooFit, arXiv:1910.14654.

[6] G. D'Agostini, A Multidimensional unfolding method based on Bayes' theorem, Nucl. Instrum. Methods Phys. Res., Sect. A **362**, 487 (1995).

[7] A. Hocker and V. Kartvelishvili, SVD approach to data unfolding, Nucl. Instrum. Methods Phys. Res., Sect. A **372**, 469 (1996).

[8] S. Schmitt, TUnfold: An algorithm for correcting migration effects in high energy physics, J. Instrum. **7**, T10003 (2012).

[9] N. D. Gagunashvili, Machine learning approach to inverse problem and unfolding procedure, arXiv:1004.2006.

[10] A. Glazov, Machine learning as an instrument for data unfolding, arXiv:1712.01814.

[11] K. Datta, D. Kar, and D. Roy, Unfolding with generative adversarial networks, arXiv:1806.00433.

[12] G. Zech and B. Aslan, Binning-free unfolding based on Monte Carlo migration, eConf **C030908**, TUGT001 (2003).

[13] L. Lindemann and G. Zech, Unfolding by weighting Monte Carlo events, Nucl. Instrum. Methods Phys. Res., Sect. A **354**, 516 (1995).

[14] D. Martschei, M. Feindt, S. Honc, and J. Wagner-Kuhr, Advanced event reweighting using multivariate analysis, J. Phys. Conf. Ser. **368**, 012028 (2012).

[15] A. Rogozhnikov, Reweighting with boosted decision trees, J. Phys. Conf. Ser. **762**, 012036 (2016).

[16] R. Aaij *et al.* (LHCb Collaboration), Observation of the Decays $\Lambda_b^0 \to \chi_{c1} p K^-$ and $\Lambda_b^0 \to \chi_{c2} p K^-$, Phys. Rev. Lett. **119**, 062001 (2017).

[17] M. Aaboud *et al.* (ATLAS Collaboration), Search for pair production of Higgsinos in final states with at least three b-tagged jets in $\sqrt{s} = 13$ TeV pp collisions using the ATLAS detector, Phys. Rev. D **98**, 092002 (2018).

[18] A. Andreassen, I. Feige, C. Frye, and M. D. Schwartz, JUNIPR: A framework for unsupervised machine learning in particle physics, Eur. Phys. J. C **79**, 102 (2019).

[19] A. Andreassen and B. Nachman, Neural networks for full phase-space reweighting and parameter tuning, arXiv:1907.08209.

[20] L. B. Lucy, An iterative technique for the rectification of observed distributions, Astron. J. **79**, 745 (1974).

[21] W. H. Richardson, Bayesian-based iterative method of image restoration, J. Opt. Soc. Am. **62**, 55 (1972).

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

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

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

[25] J. Brehmer, G. Louppe, J. Pavez, and K. Cranmer, Mining gold from implicit models to improve likelihood-free inference, arXiv:1805.12244.

[26] E. Bothmann and L. Del Debbio, Reweighting a parton shower using a neural network: The final-state case, J. High Energy Phys. 01 (2019) 033.

[27] L. A. Shepp and Y. Vardi, Maximum likelihood reconstruction for emission tomography, IEEE Trans. Med. Imaging **1**, 113 (1982).

[28] A. P. Dempster, N. M. Laird, and D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. R. Stat. Soc. Ser. B **39**, 1 (1977).

[29] C. F. J. Wu, On the convergence properties of the EM algorithm, Ann. Stat. **11**, 95 (1983).

[30] Y. Vardi, L. A. Shepp, and L. Kaufman, A statistical model for positron emission tomography, J. Am. Stat. Assoc. **80**, 8 (1985).

[31] M. Kuusela, Statistical issues in unfolding methods for high energy physics, Aalto University, Master's thesis, 2012.

[32] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevLett.124.182001 for "a discussion and derivation of OmniFold and IBU from a maximum likelihood formulation.

[33] R. Kogler *et al.*, Jet substructure at the large Hadron collider: Experimental review, Rev. Mod. Phys. **91**, 045003 (2019).

[34] M. Bahr *et al.*, Herwig++ physics and manual, Eur. Phys. J. C **58**, 639 (2008).

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

[36] J. Bellm *et al.*, Herwig 7.1 release note, arXiv:1705.06919.

[37] ATLAS Run 1 PYTHIA8 tunes, Technical Report No. ATL-PHYS-PUB-2014-021, CERN, Geneva, 2014.

[38] T. Sjöstrand, S. Mrenna, and P. Z. Skands, A brief introduction to PYTHIA 8.1, Comput. Phys. Commun. **178**, 852 (2008).

[39] T. Sjöstrand, S. Mrenna, and P. Z. Skands, PYTHIA 6.4 physics and manual, J. High Energy Phys. 05 (2006) 026.

[40] T. Sjöstrand, S. Ask, J. R. Christiansen, R. Corke, N. Desai, P. Ilten, S. Mrenna, S. Prestel, C. O. Rasmussen, and P. Z. Skands, An introduction to PYTHIA 8.2, Comput. Phys. Commun. **191**, 159 (2015).

[41] J. de Favereau, C. Delaere, P. Demin, A. Giammanco, V. Lematre, 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.

[42] M. Cacciari, G. P. Salam, and G. Soyez, The anti-k(t) jet clustering algorithm, J. High Energy Phys. 04 (2008) 063.

[43] M. Cacciari, G. P. Salam, and G. Soyez, FastJet user manual, Eur. Phys. J. C **72**, 1896 (2012).

[44] M. Cacciari and G. P. Salam, Dispelling the N^3 myth for the k_t jet-finder, Phys. Lett. B **641**, 57 (2006).

[45] P. T. Komiske, E. M. Metodiev, and J. Thaler, Energy flow networks: Deep sets for particle jets, J. High Energy Phys. 01 (2019) 121.

[46] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Póczos, R. Salakhutdinov, and A. J. Smola, Deep sets, in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA* (Curran Associates, Inc., Red Hook, 2017), pp. 3391–3401.

[47] M. Tanabashi *et al.* (Particle Data Group), Review of particle physics, Phys. Rev. D **98**, 030001 (2018).

[48] P. T. Komiske and E. M. Metodiev, EnergyFlow, https://energyflow.network/.

[49] F. Chollet, Keras, https://github.com/fchollet/keras (2017).

[50] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, Tensorflow: A system for large-scale machine learning, in *OSDI* (USENIX, Berkeley, 2016), Vol. 16, pp. 265–283.

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

[52] A. J. Larkoski, I. Moult, and B. Nachman, Jet substructure at the large Hadron collider: A review of recent advances in theory and machine learning, arXiv:1709.04464.

[53] J. Thaler and K. Van Tilburg, Identifying boosted objects with N-subjettiness, J. High Energy Phys. 03 (2011) 015.

[54] J. Thaler and K. Van Tilburg, Maximizing Boosted Top Identification by Minimizing N-subjettiness, J. High Energy Phys. 02 (2012) 093.

[55] D. Krohn, J. Thaler, and L.-T. Wang, Jet trimming, J. High Energy Phys. 02 (2010) 084.

[56] S. D. Ellis, C. K. Vermilion, and J. R. Walsh, Recombination algorithms and jet substructure: Pruning as a tool for heavy particle searches, Phys. Rev. D **81**, 094023 (2010).

[57] S. D. Ellis, C. K. Vermilion, and J. R. Walsh, Techniques for improved heavy particle searches with jet substructure, Phys. Rev. D **80**, 051501(R) (2009).

[58] M. Dasgupta, A. Fregoso, S. Marzani, and G. P. Salam, Towards an understanding of jet substructure, J. High Energy Phys. 09 (2013) 029.

[59] A. J. Larkoski, S. Marzani, G. Soyez, and J. Thaler, Soft drop, J. High Energy Phys. 05 (2014) 146.

[60] FastJet Contrib, https://fastjet.hepforge.org/contrib/.

[61] B. Nachman and C. Shimmin, AI safety for high energy physics, arXiv:1910.08606.

[62] B. Nachman, A guide for deploying deep learning in LHC searches: How to achieve optimality and account for uncertainty, arXiv:1909.03081.

[63] F. Topsoe, Some inequalities for information divergence and related measures of discrimination, IEEE Trans. Inf. Theory **46**, 1602 (2000).

[64] P. Gras, S. Hche, D. Kar, A. Larkoski, L. Lnnblad, S. Pltzer, A. Sidmok, P. Skands, G. Soyez, and J. Thaler, Systematics of quark/gluon tagging, J. High Energy Phys. 07 (2017) 091.

[65] S. Bright-Thonney and B. Nachman, Investigating the Topology Dependence of Quark and Gluon Jets, J. High Energy Phys. 03 (2019) 098.

[66] P. T. Komiske, E. M. Metodiev, and J. Thaler, Metric Space of Collider Events, Phys. Rev. Lett. **123**, 041801 (2019).

[67] P. T. Komiske, R. Mastandrea, E. M. Metodiev, P. Naik, and J. Thaler, Exploring the space of jets with CMS open data, Phys. Rev. D **101**, 034009 (2020).

[68] B. Nachman, M. Urbanek, W. A. de Jong, and C. Bauer, Unfolding quantum computer readout noise, arXiv:1910.01969.

[69] A. Andreassen, P. T. Komiske, E. M. Metodiev, B. Nachman, and J. Thaler, OmniFold, https://github.com/ericmetodiev/OmniFold (2019).

[70] A. Andreassen, P. Komiske, E. Metodiev, B. Nachman, and J. Thaler, PYTHIA/HERWIG + DELPHES jet datasets for omnifold unfolding, zenodo, https://doi.org/10.5281/zenodo.3548091 (2019).

[71] X. Chen *et al.*, Open is not enough, Nat. Phys. **15**, 113 (2019).