# Learning Uncertainties the Frequentist Way: Calibration and Correlation in 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 25 July 2022; published 15 August 2022)

Calibration is a common experimental physics problem, whose goal is to infer the value and uncertainty of an unobservable quantity $Z$ given a measured quantity $X$. Additionally, one would like to quantify the extent to which $X$ and $Z$ are correlated. In this Letter, we present a machine learning framework for performing frequentist maximum likelihood inference with Gaussian uncertainty estimation, which also quantifies the mutual information between the unobservable and measured quantities. This framework uses the Donsker-Varadhan representation of the Kullback-Leibler divergence—parametrized with a novel Gaussian ansatz—to enable a simultaneous extraction of the maximum likelihood values, uncertainties, and mutual information in a single training. We demonstrate our framework by extracting jet energy corrections and resolution factors from a simulation of the CMS detector at the Large Hadron Collider. By leveraging the high-dimensional feature space inside jets, we improve upon the nominal CMS jet resolution by upward of 15%.

DOI: 10.1103/PhysRevLett.129.082001

One of the most foundational tasks in high energy physics (HEP) is the inference of an unobservable quantity given a measured quantity, which is often referred to as calibration. For example, the kinematic properties of a given particle must be reconstructed from signatures registered in various detector elements. This inference task can be challenging when the reconstruction requires high-dimensional inputs. Machine learning (ML) is a natural tool for performing high-dimensional reconstruction, and there has been significant progress in utilizing the ML method for estimating the energies of various objects, including photons [1], muons [2], single hadrons [3–8], and sprays of hadrons (jets) [9–19] at colliders, kinematic reconstruction in deep inelastic scattering [20,21], and neutrino energies in a variety of experiments [22–27]. Further ideas can be found in Ref. [28].

Abstractly, the calibration task can be described as quantifying the relationship between two random variables $X \in \mathbb{R}^M$ and $Z \in \mathbb{R}^N$. Here, $X$ is the measured quantity and $Z$ is the unobservable ("latent") quantity (throughout this Letter, uppercase letters represent random variables and lowercase letters represent realizations of those random variables). A reconstruction technique produces a function $\hat{z}: \mathbb{R}^M \to \mathbb{R}^N$, which is determined by minimizing a loss functional over sample data (real or synthetic). While ML methods are effective even when $M$ and $N$ are large, most existing methods have the undesirable property of being prior dependent [29]. This means that $\hat{z}$ depends on the probability density $p(z)$ used during training. As a result, the calibration is not universal and caution must be taken when applying it to different event samples. Furthermore, some calibration methods simply produce a point estimate, with no estimation of the corresponding uncertainty. In the HEP context, this uncertainty is usually called the resolution. Quantifying the reconstruction resolution is relevant for a variety of purposes, including the computation of significance variables [30,31] and background estimation [32,33]. Various ML approaches for resolution determination have been recently studied for HEP [34–40], but they typically require additional training or model complexity. See Ref. [41] for a complementary approach to frequentist inference.

In this Letter, we introduce a simple ML framework for calibration that simultaneously estimates the following quantities: (1) a prior-independent maximum-likelihood calibration $\hat{z}(x) = \text{argmax}_z p(x|z)$, (2) a Gaussian resolution around $\hat{z}(x)$, $\hat{\sigma}_z(x)$, (3) the log-likelihood ratio $\log[p(x|z)/p(x)]$, and (4) the mutual information between $X$ and $Z$, $I(X;Z)$. To extract $\hat{z}(x)$ and $\hat{\sigma}_z(x)$ in a single training, we use a novel Gaussian ansatz to parametrize the log-likelihood ratio with an interpretable network

architecture. Mutual information is a powerful statistic for quantifying the (nonlinear) correlation between two random variables, and it appears due to our choice of loss function. After describing the Gaussian ansatz construction, we illustrate the above features in a case study involving jet reconstruction at the Large Hadron Collider (LHC).

Our calibration method builds upon the Mutual Information Neural Estimator (MINE) introduced in Ref. [42]. With MINE, the Donsker-Varadhan representation [43] of the Kullback-Leibler divergence [44] is used to estimate $I(X;Z)$ by training a network to minimize a particular loss functional. We first show that a network minimizing this loss functional yields the likelihood $p(x|z)$, which in principle contains all the information necessary for frequentist inference. Performing this inference in practice, though, involves difficult optimization tasks, which are even more difficult if one wants to extract the resolution. With the Gaussian ansatz, we parametrize the MINE network such that the inferred value, and especially the resolution, are easy to extract after ML training.

The starting point for our calibration method is the concept of mutual information (MI) defined as

$$I(X;Z) = \int dx\, dz\, p(x,z) \log \frac{p(x,z)}{p(x)p(z)}, \tag{1}$$

where $p$ denotes the probability density of the respective random variable. This equation has the property that $I(X;Z) = 0$ if and only if $X$ and $Z$ are independent, which is equivalent to $p(x,z) = p(x)p(z)$. Therefore, the MI quantifies the interdependence between $X$ and $Z$.

The MI is a special case of the Kullback-Leibler (KL) divergence $D_{\mathrm{KL}}(P\|Q)$ when $P = P_{XZ}$ is the joint probability distribution of $X$ and $Z$ [i.e., $p(x,z)$], and $Q = P_X \otimes P_Z$ is the product of the marginals [i.e., $p(x)p(z)$]. It is well known that the KL divergence can be cast in the Donsker-Varadhan (DV) representation [45]:

$$D_{\mathrm{KL}}(P\|Q) = \sup_{T\in\mathcal{T}} \{\mathbb{E}_P[T] - \log(\mathbb{E}_Q[e^T])\}, \tag{2}$$

where $\mathbb{E}_\bullet$ represents the expectation value over probability density $\bullet$, and the supremum is over the space of functions $\mathcal{T}$ such that both expectations are finite.

Following the MINE construction in Ref. [42], we use the DV representation to build an estimator for the mutual information from a finite dataset. For functions $T: \mathbb{R}^M \times \mathbb{R}^N \to \mathbb{R}$, we can place a lower bound on $I(X;Z)$ by minimizing a loss functional $\mathcal{L}_{\mathrm{DVR}}$ over $T \in \mathcal{T}$:

$$I(X;Z) \geq -\inf_{T\in\mathcal{T}} \mathcal{L}_{\mathrm{DVR}}[T], \tag{3}$$

where the DV representation (DVR) loss is

$$\mathcal{L}_{\mathrm{DVR}}[T] = -(\mathbb{E}_{P_{XZ}}[T] - \log(\mathbb{E}_{P_X\otimes P_Z}[e^T])). \tag{4}$$

Given a finite dataset of $(x,z)$ pairs, the expectations in Eq. (4) can be estimated from sample averages. To estimate the second term, one can simply shuffle the $x$'s and $z$'s, as done in Ref. [42]. The space of functions $\mathcal{T}$ can be parametrized by neural networks, in which case the DVR loss functional can be minimized using standard gradient descent. As long as $\mathcal{T}$ is sufficiently expressive, the bound in Eq. (3) will be saturated, so the minimum loss is an estimate of $-I(X;Z)$ [46].

Taking the functional derivative of the DVR loss functional with respect to $T$, we see that the supremum of $\mathcal{L}[T]$ is obtained when

$$T(x,z) = \log \frac{p(x|z)}{p(x)} + c, \tag{5}$$

where $c$ is any constant that we can set to zero without loss of generality [in practice, we determine and then subtract $c$ numerically by noting that the second term of Eq. (4) is an estimate of $c$ in the asymptotic limit]. Therefore, if the MINE is well trained, we can use $T$ as an estimate of the log-likelihood density ratio. As with most machine learning applications, this requires that the space of neural networks $\mathcal{T}$ is sufficiently expressive, that there are enough training data, and that the gradient descent algorithm successfully finds the minimum of Eq. (4). Given this, we can then perform maximum likelihood inference given $x$:

$$\hat{z}(x) = \mathrm{argmax}_z T(x,z). \tag{6}$$

Crucially, this inference strategy for $z$ is independent of the prior $p(z)$, which is a property desirable for calibration tasks. Unlike for standard regression [29], the learned estimate $\hat{z}$ does not depend on the distribution of $z$ samples in the training set [if desired, one could do Bayesian inference and obtain the posterior $p(z|x)$ by adding the prior $\log p(z)$ to $T(x,z)$].

If $X$ does not contain complete information about $Z$, then there will be uncertainty in our inference of $z$. Assuming the likelihood $p(x|z)$ is well approximated by a Gaussian density, the uncertainty in the inference is given by the covariance matrix:

$$[\hat{\sigma}_z^2(x)]_{ij} = -\left[\frac{\partial^2 T(x,z)}{\partial z_i \partial z_j}\right]^{-1}\Bigg|_{z=\hat{z}}, \tag{7}$$

which is again prior independent.

So far, we have shown that the MINE network can be used to perform frequentist inference. While $T$ itself depends on the prior $p(z)$, the inference $\hat{z}$ and resolution $\hat{\sigma}_z$ do not. However, both the maximum likelihood estimate in Eq. (6) and the local resolution in Eq. (7) are difficult to evaluate numerically. In the case of maximization, the learned $T$

may be highly nonconvex, and the true maxima difficult to find using gradient descent. In the case of the second derivative, its evaluation is numerically sensitive to the choice of activation function in the MINE network. For example, if one uses the common rectified linear unit (ReLU) activation function or its variants, then all analytic second derivatives of the network are zero.

In order to facilitate a numerical estimate of the maximum likelihood and local resolution, we introduce the following Gaussian ansatz parametrization for $T$:

$$T(x,z) = A(x) + [z - B(x)]D(x) + \frac{1}{2}[z - B(x)]^T C(x,z)[z - B(x)], \quad (8)$$

where $A:\mathbb{R}^N \to \mathbb{R}$, $B:\mathbb{R}^N \to \mathbb{R}^M$, $C:\mathbb{R}^N \times \mathbb{R}^M \to \mathrm{Sym}(M,\mathbb{R})$, and $D:\mathbb{R}^N \to \mathbb{R}^M$ are each neural networks. We call this the Gaussian ansatz, since it resembles the logarithm of a Gaussian likelihood density. Unlike a Gaussian likelihood, though, the Gaussian ansatz is highly expressive, and is in fact a universal function approximator. Specifically, any function $f(x,z)$ that admits a Taylor expansion in $z$ around $B(x)$ can be expanded in this form. The functions $A(x)$ and $D(x)$ capture the zeroth and first order dependences of $f$ on $z$, respectively. The function $C(x,z)$ captures any quadratic or higher dependence of the Taylor expansion of $f$ on $z$.

The Gaussian ansatz enables an elegant strategy to extract Eqs. (6) and (7). Since the optimal $T(x,z)$ is bounded from above, we can take $D(x)$ to be everywhere zero without loss of expressivity. In this case, $T$ will achieve critical values at $z = B(x)$. Moreover, if $C(x,B(x)) < 0$, then these critical values will be (local) likelihood maxima:

$$\hat{z}(x) = B(x). \quad (9)$$

While the Gaussian ansatz does not necessarily protect against local maxima, it does yield a numerical estimate of the local resolution:

$$\hat{\sigma}_z^2(x) = -[C(x,B(x))]^{-1}. \quad (10)$$

Moreover, the (negative) loss of the Gaussian ansatz with respect to the functional in Eq. (4) will be a lower bound for the mutual information $I(X;Z)$, which is saturated in the asymptotic limit of an infinitely large network with infinite data.

The Gaussian ansatz is therefore capable of estimating—from a single dataset of $(x,z)$ pairs and a single training—the maximum likelihood inferred value of $z$ given $x$, the local resolution on that inference, and the mutual information between $X$ and $Z$. This can be achieved without having to perform any additional optimization problems, derivative estimations, or postprocessing beyond the single matrix inversion in Eq. (10). In practice, we find it convenient to start the training with nonzero $D(x)$ to aid the convergence of the model, and then numerically force $D \to 0$ through an increasing $L_1$ regularization. This helps the model achieve a global, rather than local, minimum.

We now demonstrate the Gaussian ansatz on an experimental collider physics task: determining jet energy corrections (JECs) and resolutions (JERs) [53]. (At the LHC, one typically calibrates transverse momenta $p_T$ instead of energies, but the terms JECs and JERs are still used.) Jets are collimated sprays of particles that are produced ubiquitously in high energy collisions. One does not have access to the "true" jet energy, however, because its constituent particles are filtered through a complicated and nonlinear detector response.

Assuming one has a good detector model, though, one can generate truth-level quantities (GEN, corresponding to $Z$) and then simulate the detector response (SIM, corresponding to $X$). Performing a simulation-based calibration, one can infer the true jet energy from a set of measured particle momenta in a jet. The multiplicative JEC factor is then defined such that the inferred jet momenta is

$$\hat{p}_T \equiv \mathrm{JEC} \times p_{T,\mathrm{SIM}} \approx p_{T,\mathrm{GEN}}. \quad (11)$$

JEC factors are often further refined through a data-based calibration using well-understood control samples, though this is separate from the procedure considered here. The JER factor arises because the inferred and generated $p_T$ values in Eq. (11) are not identical. The JER is typically expressed as a fractional quantity:

$$\hat{\sigma}_{p_T} = \mathrm{JER} \times p_{T,\mathrm{SIM}}. \quad (12)$$

In the language of statistics, the JER is a type of "uncertainty," since it represents the limited information about $Z$ contained in $X$. In the HEP context, though, this quantity is instead called a "resolution"; see Ref. [29] for further discussion.

The JEC factor is a function of the measured quantities, primarily the detector-level jet $p_T$ and pseudorapidity $\eta$. The JEC can be obtained from fits to simulation [54–58] using a technique called numerical inversion [59]. The JER can be also determined in simulation by fitting the peak region of the detector response $\hat{p}_T/p_{T,\mathrm{SIM}}$ to a Gaussian distribution. Here, we consider an alternate (and arguably simpler) approach to JEC and JER extraction.

For our case study, we use the Gaussian ansatz to improve upon the JEC factors provided by the CMS experiment in their 2011 public data release [60]. We use the same 2011 CMS Open Simulation [61] samples as in Ref. [62], which are based on dijets generated in PYTHIA6 [63] with a GEANT4-based [64] simulation of the CMS detector. This dataset was translated from the original CMS analysis object data ROOT-based format into an easier-to-use MIT Open Data HDF5 format [65]. Each

SIM event consists of a list of particle flow candidates (PFCs), which are the reconstructed four-momentum and particle identification (PID) for each measured particle. The PFCs are clustered into jets, using the anti-$k_t$ jet algorithm with $R = 0.5$ [66–68]. For each jet, truth-level GEN jet information is also provided, as well as the CMS-prescribed JEC. CMS-prescribed JERs are estimated using Ref. [53].

We select jets whose GEN transverse momentum is in the range $p_T \in [500, 1000]$ GeV. The lower bound of 500 GeV is to avoid any turn-on effects due to the $p_{T,\mathrm{SIM}} > 375$ GeV cut applied to the dataset as a whole. We require that the GEN pseudorapidity satisfies $|\eta| < 2.4$, and that jets are at least "medium" jet quality [69]. The latent variable of interest is $Z = p_{T,\mathrm{GEN}}$, and the measured quantity $X = X_{\mathrm{SIM}}$ depends on the choice of ML architecture. All momenta are divided by a fixed scale of 1000 GeV, so that the data values are roughly $\mathcal{O}(1)$. In total, $5 \times 10^6$ jets are used for training across the whole $p_T \in [500, 1000]$ GeV range.

We consider four different ML models of increasing sophistication. (1) Dense neural network (DNN): The input $X$ consists only of the overall jet kinematic properties, with $X = (p_T, \eta, \phi)_{\mathrm{SIM}}$, which is the same information used in the CMS calibration procedure in Ref. [53]. Each of the functions $A$, $B$, $C$, and $D$ are constructed as fully connected neural networks, with three hidden layers of size 64 and ReLU activations. (2) Energy flow network (EFN): The input $X$ consists of the entire set of PFC three-momenta from the jet. Each of the functions $A$, $B$, $C$, and $D$ are constructed as energy EFNs [70]. EFNs are permutation-invariant functions of point clouds inspired by the deep sets formalism [71]. They take the form $f(\{\vec{p}_i\}) = F(\sum_i p_{Ti}\Phi(\eta_i, \phi_i))$, which exhibits manifest infrared and colinear (IRC) safety. For each EFN, the $\Phi$ and $F$ functions consist of three hidden layers of respective sizes (50,50,64) with ReLU activations. Since $C$ is a function of both $X$ and $Z$, the $Z$ is appended as an input to the $F$ function. (3) Particle flow network (PFN): The same input features as the EFN but inserted into a PFN [70,71], which does not impose IRC safety. PFNs take the form $f(\{\vec{p}_i\}) = F(\sum_i \Phi(p_{T_i}, \eta_i, \phi_i))$. (4) PFN PID: The same as the PFN model, but in addition to the three-momenta of each PFC, the reconstructed PID is included as an input feature. We follow the PID labeling scheme of Ref. [70] for photon, charged hadron, etc. Each of these models is trained for 200 epochs using the ADAM optimizer [72], with a learning rate of $\alpha = 10^{-4}$ and a batch size of 2048. All model parameters are given an $L_2$ regularization loss with weight $\lambda_2 = 10^{-6}$. The $D$ network is given an overall $L_1$ regularization loss of $\lambda_D = 10^{-3}$ to slowly force it to zero by the end of the training. Every 50 epochs, $\alpha$ is reduced by a factor of 5, and $\lambda_D$ is increased by a factor of 10. To aid the numerical convergence, each model is pretrained with a mean-squared-error loss $\mathcal{L}_{\mathrm{pre}}[B,C] = \mathbb{E}_{P_{XZ}}[(B(x) - z)^2 + (C(x,z) + \mathrm{cov}(X,Z))^2]$.

TABLE I. Gaussian ansatz results for the four ML models compared to the CMS 2011 baseline [53]. On a test dataset of GEN jets with $p_T \in [695, 705]$ GeV, we show the inferred $\hat{p}_T$, its resolution $\hat{\sigma}_{p_T}$, and the learned mutual information between $X = X_{\mathrm{SIM}}$ and $Z = p_{T,\mathrm{GEN}}$. The $\pm$ values correspond to the standard deviation of the $\hat{p}_T$ and $\hat{\sigma}_{p_T}$ distributions across the test set, and bold face indicates the best resolution and highest mutual information.

| Model | Mean $\hat{p}_T$ (GeV) | Mean $\hat{\sigma}_{p_T}$ (GeV) | $I(X;Z)$ |
|---|---|---|---|
| DNN | $698 \pm 37.7$ | $35.7 \pm 2.1$ | 1.23 |
| EFN | $695 \pm 37.3$ | $32.6 \pm 2.3$ | 1.26 |
| PFN | $697 \pm 36.9$ | $32.5 \pm 2.5$ | 1.27 |
| PFN PID | $695 \pm 35.1$ | $\mathbf{30.8 \pm 3.6}$ | **1.32** |
| CMS 2011 | $695 \pm 38.4$ | $36.9 \pm 1.7$ | ⋯ |

In Table I, we show the results of the training in a narrow bin of $p_{T,\mathrm{GEN}} \in [695, 705]$ GeV. If our models yield unbiased estimators of the GEN $p_T$, then the inferred $\hat{p}_T$ distribution should be centered near 700 GeV, which it is for all models. Adding more information to the model should not decrease the mutual information, and if useful, that information should improve the resolution. We see indeed that the resolution improves with increasing model sophistication, as does the mutual information $I(X;Z)$. The resolution from the DNN, which uses the same information as the CMS procedure, is marginally better than the nominal CMS 2011 jet resolution from Ref. [53]. The PFN PID model exhibits the best resolution, which is roughly 15% better on average than the CMS baseline.

In Fig. 1, we show the distribution of $\hat{\sigma}_{p_T}$ in the same $p_{T,\mathrm{GEN}} \in [695, 705]$ GeV bin. As the model sophistication increases, the resolution increases (i.e., the $\hat{\sigma}_{p_T}$ shift downward). The non-Gaussian behavior of the ML models

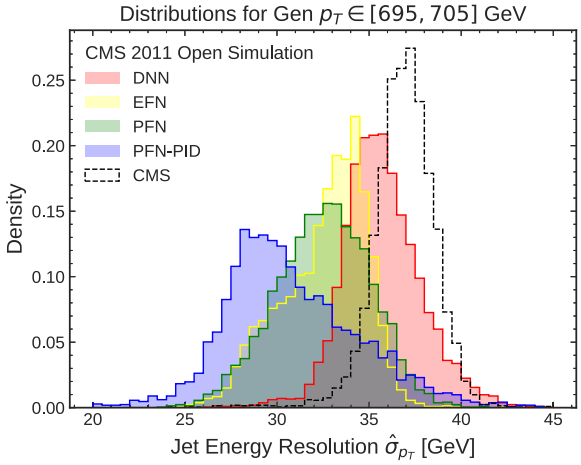


FIG. 1. Learned JER distribution for the four models compared to the CMS 2011 baseline. The dataset is the same as in Table I. On average, the PFN PID exhibits 15% better resolution (i.e., smaller values) than the CMS default.

is expected, since these models are exploiting additional information beyond the $p_T$. In principle, the resolution should never degrade by adding more information, but we do find a long right tail for the PFN PID model due to incomplete ML convergence (we verified that the tail shrinks and the resolution improves with increasing training statistics, but we were limited by machine memory considerations). We conclude that the measured PFC momenta, along with the PIDs, contain useful information for jet energy calibration that is lost when only considering the total jet momentum.

In this Letter, we presented an extension of the MINE framework, the Gaussian ansatz, capable of simultaneously performing frequentist inference, extracting Gaussian uncertainties, and quantifying mutual information between random variables. All of these tasks were performed in a single training, with no additional postprocessing. Using this ML framework, we were able to take advantage of the full jet particle information in the CMS Open Simulation to improve the measured jet resolution by approximately 15%. Studies by the ATLAS Collaboration have used sequential calibration on a handful of observables to improve their resolution [56–58], and the Gaussian ansatz may allow for further improvements by allowing for simultaneous calibrations of any number of input features. We look forward to further developments in ML-based calibration and correlations methods in HEP and beyond.

The code for the general-use Gaussian ansatz framework can be found at Ref. [73]. The code and data for the jet energy calibration study, in particular, are available at [74].

We would like to thank Patrick Komiske for helpful discussions about EFNs and PFNs, Govert Nijs for helpful discussions on numerics and convergence, and Jennifer Roloff for helpful discussions about jet calibrations. We are grateful to Phiala Shanahan and Andrew Pochinsky for providing access to the Wombat cluster for some of the calculations undertaken in this work. R. G. and J. T. are supported by the National Science Foundation under Cooperative Agreement No. PHY-2019786 (The NSF AI Institute for Artificial Intelligence and Fundamental Interactions), and by the U.S. DOE Office of High Energy Physics under Grant No. DE-SC0012567. B. N. is supported by the U.S. Department of Energy, Office of Science under Contract No. DE-AC02-05CH11231.

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

[1] Albert M. Sirunyan *et al.* (CMS Collaboration), Electron and photon reconstruction and identification with the CMS experiment at the CERN LHC, J. Instrum. **16**, P05014 (2021).
[2] 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).
[3] Dawit Belayneh *et al.*, Calorimetry with deep learning: Particle simulation and reconstruction for collider physics, Eur. Phys. J. C **80**, 688 (2020).
[4] ATLAS Collaboration, Deep learning for pion identification and energy calibration with the ATLAS detector, Report No. ATL-PHYS-PUB-2020-018, 2020, https://cds.cern.ch/record/2724632.
[5] 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).
[6] 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.
[7] L. Polson, L. Kurchaninov, and M. Lefebvre, Energy reconstruction in a liquid argon calorimeter cell using convolutional neural networks, J. Instrum. **17**, P01002 (2022).
[8] 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).
[9] ATLAS Collaboration, Generalized numerical inversion: A neural network approach to jet calibration, Report No. ATL-PHYS-PUB-2018-013, 2018, https://cds.cern.ch/record/2630972.
[10] ATLAS Collaboration, Simultaneous jet energy and mass calibrations with neural networks, Report No. ATL-PHYS-PUB-2020-001, 2020, https://cds.cern.ch/record/2706189.
[11] 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).
[12] Rüdiger Haake and Constantin Loizides, Machine learning based jet momentum reconstruction in heavy-ion collisions, Phys. Rev. C **99**, 064904 (2019).
[13] 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.
[14] 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.
[15] 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.
[16] ATLAS Collaboration, Convolutional neural networks with event images for pileup mitigation with the ATLAS detector, CERN Technical Report No. ATL-PHYS-PUB-2019-028, 2019, https://cds.cern.ch/record/2684070.
[17] Benedikt Maier, Siddharth M. Narayanan, Gianfranco de Castro, Maxim Goncharov, Christoph Paus, and Matthias Schott, Pile-up mitigation using attention, Mach. Learn. Sci. Technol. **3**, 025012 (2022).
[18] Gregor Kasieczka, Michel Luchmann, Florian Otterpohl, and Tilman Plehn, Per-object systematics using deep-learned calibration, SciPost Phys. **9**, 089 (2020).

[19] 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).

[20] Markus Diefenthaler, Abduhhal Farhat, Andrii Verbytskyi, and Yuesheng Xu, Deeply learning deep inelastic scattering kinematics, arXiv:2108.11638.

[21] Miguel Arratia, Daniel Britzger, Owen Long, and Benjamin Nachman, Reconstructing the kinematics of deep inelastic scattering with deep learning, Nucl. Instrum. Methods Phys. Res., Sect. A **1025**, 166164 (2022).

[22] 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.

[23] S. Delaquis *et al.* (EXO Collaboration), Deep neural networks for energy and position reconstruction in EXO-200, J. Instrum. **13**, P08023 (2018).

[24] Pierre Baldi, Jianming Bian, Lars Hertel, and Lingge Li, Improved energy reconstruction in NO($\nu$)A with regression convolutional neural networks, Phys. Rev. D **99**, 012011 (2019).

[25] R. Abbasi *et al.*, A convolutional neural network based cascade reconstruction for the IceCube neutrino observatory, J. Instrum. **16**, P07041 (2021).

[26] M. G. Aartsen *et al.* (IceCube Collaboration), Cosmic ray spectrum from 250 TeV to 10 PeV using IceTop, Phys. Rev. D **102**, 122001 (2020).

[27] 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).

[28] Matthew Feickert and Benjamin Nachman, A living review of machine learning for particle physics, arXiv:2102.02770.

[29] Rikab Gambhir, Benjamin Nachman, and Jesse Thaler, companion paper, Bias and priors in machine learning calibrations for high energy physics, Phys. Rev. D **106**, 036011 (2022).

[30] Albert M. Sirunyan *et al.* (CMS Collaboration), Performance of missing transverse momentum reconstruction in proton-proton collisions at $\sqrt{s} = 13$ TeV using the CMS detector, J. Instrum. **14**, P07004 (2019).

[31] Benjamin Nachman and Christopher G. Lester, Significance variables, Phys. Rev. D **88**, 075013 (2013).

[32] Georges Aad *et al.* (ATLAS Collaboration), Search for squarks and gluinos with the ATLAS detector in final states with jets and missing transverse momentum using 4.7 fb$^{-1}$ of $\sqrt{s} = 7$ TeV proton-proton collision data, Phys. Rev. D **87**, 012008 (2013).

[33] Georges Aad *et al.* (ATLAS Collaboration), Search for new phenomena in events with an energetic jet and missing transverse momentum in $pp$ collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector, Phys. Rev. D **103**, 112006 (2021).

[34] 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).

[35] Sanha Cheong, Aviv Cukierman, Benjamin Nachman, Murtaza Safdari, and Ariel Schwartzman, Parametrizing the detector response with neural networks, J. Instrum. **15**, P01030 (2020).

[36] 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).

[37] Marco Bellagente, Manuel Haußmann, Michel Luchmann, and Tilman Plehn, Understanding event-generation networks via uncertainties, arXiv:2104.04543.

[38] Braden Kronheim, Michelle Kuchera, Harrison Prosper, and Alexander Karbo, Bayesian neural networks for fast SUSY predictions, Phys. Lett. B **813**, 136041 (2021).

[39] Jack Y. Araz and Michael Spannowsky, Combine and conquer: Event reconstruction with Bayesian ensemble neural networks, J. High Energy Phys. 04 (2021) 296.

[40] Braden Kronheim, Michelle P. Kuchera, Harrison B. Prosper, and Raghuram Ramanujan, Implicit quantile neural networks for jet simulation and correction, in *Proceedings of NeurIPS 2021—Thirty-fifth Workshop on Machine Learning and the Physical Sciences, Vancouver, Canada, December, 2021*, arXiv:2111.11415.

[41] Niccolò Dalmasso, David Zhao, Rafael Izbicki, and Ann B. Lee, Likelihood-free frequentist inference: Bridging classical statistics and machine learning in simulation and uncertainty quantification, arXiv:2107.03920.

[42] 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.

[43] M. D. Donsker and S. R. S. Varadhan, Asymptotic evaluation of certain Markov process expectations for large time, Comm. Pure Appl. Math. **28**, 1 (1975).

[44] Solomon Kullback and Richard A. Leibler, On information and sufficiency, Ann. Math. Stat. **22**, 79 (1951).

[45] M. D. Donsker and S. R. S. Varadhan, Asymptotic evaluation of certain Markov process expectations for large time—III, Commun. Pure Appl. Math. **29**, 389 (1976).

[46] Other loss functionals exist that are capable of providing lower bounds on MI. For example, if we write the $f$-divergence representation of the KL divergence [47,48], the corresponding loss functional is a variation of the maximum likelihood classifier (MLC) loss [49–51]:

$$\mathcal{L}_{\mathrm{MLC}}[T] = -(\mathbb{E}_{P_{XZ}}[T] - \mathbb{E}_{P_X \otimes P_Z}[e^T - 1]).$$

Numerical and analytic studies Refs. [42,52], as well as our own empirical studies, show that the DVR loss has better numerical convergence properties than the MLC loss.

[47] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka, F-GAN: Training generative neural samplers using variational divergence minimization, arXiv:1606.00709.

[48] XuanLong Nguyen, Martin J. Wainwright, and Michael I. Jordan, Estimating divergence functionals and the likelihood ratio by convex risk minimization, IEEE Trans. Inf. Theory **56**, 5847 (2010).

[49] Raffaele Tito D'Agnolo and Andrea Wulzer, Learning new physics from a machine, Phys. Rev. D **99**, 015014 (2019).

[50] Raffaele Tito D'Agnolo, Gaia Grosso, Maurizio Pierini, Andrea Wulzer, and Marco Zanetti, Learning multivariate new physics, Eur. Phys. J. C **81**, 89 (2021).

[51] Benjamin Nachman and Jesse Thaler, E pluribus unum ex machina: Learning from many collider events at once, Phys. Rev. D **103**, 116013 (2021).
[52] Avraham Ruderman, Mark Reid, Dario Garcia-Garcia, and James Petterson, Tighter variational representations of $f$-divergences via restriction to probability measures, arXiv: 1206.4664.
[53] V. Khachatryan, A. M. Sirunyan, A. Tumasyan, W. Adam, E. Asilar, T. Bergauer, J. Brandstetter, E. Brondolin, M. Dragicevic, J. Erö *et al.*, Jet energy scale and resolution in the CMS experiment in $pp$ collisions at 8 TeV, J. Instrum. **12**, P02014 (2017).
[54] Serguei Chatrchyan *et al.* CMS Collaboration, Determination of jet energy calibration and transverse momentum resolution in CMS, J. Instrum. **6**, P11002 (2011).
[55] 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).
[56] M. Aaboud *et al.* (ATLAS Collaboration), Jet energy scale measurements and their systematic uncertainties in proton-proton collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector, Phys. Rev. D **96**, 072002 (2017).
[57] Georges Aad *et al.* (ATLAS Collaboration), Jet energy measurement and its systematic uncertainty in proton-proton collisions at $\sqrt{s} = 7$ TeV with the ATLAS detector, Eur. Phys. J. C **75**, 17 (2015).
[58] Morad Aaboud *et al.* (ATLAS Collaboration), Determination of jet calibration and energy resolution in proton-proton collisions at $\sqrt{s} = 8$ TeV using the ATLAS detector, Eur. Phys. J. C **80**, 1104 (2020).
[59] Aviv Cukierman and Benjamin Nachman, Mathematical properties of numerical inversion for jet calibrations, Nucl. Instrum. Methods Phys. Res., Sect. A **858**, 1 (2017).
[60] Achintya Rao, CMS releases new batch of LHC open data (2016), https://home.cern/news/news/computing/cms-releases-new-batch-lhc-open-data.
[61] CERN open data portal, https://opendata.cern.ch.
[62] 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).
[63] Torbjörn Sjöstrand, Stephen Mrenna, and Peter Skands, PYTHIA 6.4 physics and manual, J. High Energy Phys. 05 (2006) 026.
[64] S. Agostinelli *et al.*, GEANT4–A simulation toolkit, Nucl. Instrum. Methods Phys. Res., Sect. A **506**, 250 (2003).
[65] Patrick Komiske, Radha Mastandrea, Eric Metodiev, Preksha Naik, and Jesse Thaler, CMS 2011A Open Data, Jet Primary Dataset, pT > 375 GeV, MOD HDF5 Format (2019).
[66] Matteo Cacciari and Gavin P. Salam, Dispelling the $N^3$ myth for the $k_t$ jet-finder, Phys. Lett. B **641**, 57 (2006).
[67] Matteo Cacciari, Gavin P. Salam, and Gregory Soyez, The anti-(k_t) jet clustering algorithm, J. High Energy Phys. 04 (2008) 063.
[68] Matteo Cacciari, Gavin P. Salam, and Gregory Soyez, FASTJET User Manual, Eur. Phys. J. C **72**, 1896 (2012).
[69] Jet performance in $pp$ collisions at 7 TeV (2010), https://cds.cern.ch/record/1279362.
[70] Patrick T. Komiske, Eric M. Metodiev, and Jesse Thaler, Energy flow networks: Deep sets for particle jets, J. High Energy Phys. 01 (2019) 121.
[71] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R. Salakhutdinov, and Alexander J. Smola, Deep sets, 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.
[72] Diederik P. Kingma and Jimmy Ba, Adam: A method for stochastic optimization, arXiv:1412.6980.https://github.com/rikab/GaussianAnsatz.
[73] https://github.com/rikab/GaussianAnsatzhttps://github.com/rikab/GaussianAnsatz/tree/main/JEC.
[74] https://github.com/rikab/GaussianAnsatz/tree/main/JEC.