Eur. Phys. J. C (2024) 84:776
https://doi.org/10.1140/epjc/s10052-024-13135-4

THE EUROPEAN PHYSICAL JOURNAL C

Regular Article - Experimental Physics

Designing observables for measurements with deep learning

Owen Long[1,a], **Benjamin Nachman**[2,3]

[1] Department of Physics and Astronomy, University of California, Riverside, CA 92521, USA
[2] Physics Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA
[3] Berkeley Institute for Data Science, University of California, Berkeley, CA 94720, USA

Received: 16 October 2023 / Accepted: 18 July 2024

Abstract Many analyses in particle and nuclear physics use simulations to infer fundamental, effective, or phenomenological parameters of the underlying physics models. When the inference is performed with unfolded cross sections, the observables are designed using physics intuition and heuristics. We propose to design targeted observables with machine learning. Unfolded, differential cross sections in a neural network output contain the most information about parameters of interest and can be well-measured by construction. The networks are trained using a custom loss function that rewards outputs that are sensitive to the parameter(s) of interest while simultaneously penalizing outputs that are different between particle-level and detector-level (to minimize detector distortions). We demonstrate this idea in simulation using two physics models for inclusive measurements in deep inelastic scattering. We find that the new approach is more sensitive than classical observables at distinguishing the two models and also has a reduced unfolding uncertainty due to the reduced detector distortions.

Contents

1 Introduction

Simulations are widely used for parameter estimation in particle and nuclear physics. A typical analysis will follow one of two paths: forward-folding or unfolding. In the forward-folding pipeline, the target physics model must be specified at the time of inference. We focus on unfolding, where detector distortions are corrected in a first step (unfolding) and then the resulting cross section can be analyzed in the context of many models by any end users. In the unfolding pipeline, the first step is to identify observables sensitive to a given parameter(s). These are typically identified using physical reasoning. Then, the differential cross sections of these observables are measured, which includes unfolding with uncertainty quantification. Finally, the measured cross sections are fit to simulation templates with different values of the target parameters. This approach has been deployed to measure fundamental parameters like the top quark mass [1] and the strong coupling constant $\alpha_s(m_Z)$ [2,3] as well as parton distribution functions [4–7] and effective or phenomenological parameters in parton shower Monte Carlo programs [8].

A key drawback of the standard pipeline is that the observables are constructed manually. There is no guarantee that the observables are maximally sensitive to the target parameters. Additionally, the observables are usually chosen based on particle-level information alone and so detector distortions may not be small. Such distortions can reduce the sensitivity to the target parameter once they are corrected for by unfolding. In some cases, the particle-level observable must be chosen manually because it must be calculable precisely in perturbation theory; this is usually not the case when Monte Carlo simulations are used for the entire statistical analysis. There have been proposals to optimize the detector-level observable for a given particle-level observable [9] since they do not need to be the same. Alternatively, one could measure

[a] e-mail: owen.long@ucr.edu (corresponding author)

Published online: 04 August 2024

the full phase space and project out the desired observable after the fact [10–29] (see Ref. [30] for an overview).

We propose to use machine learning for designing observables that are maximally sensitive to a given parameter(s) or model discrimination while also being minimally sensitive to detector distortions. Simultaneous optimization ensures that we only use regions of phase space that are measurable. A tailored loss function is used to train neural networks. We envision that this approach could be used for any case where simulations are used for parameter estimation. For concreteness, we demonstrate the new technique to the case of differentiating two parton shower Monte Carlo models of deep inelastic scattering. While neither model is expected to match data exactly, the availability of many events with corresponding detailed simulations makes this a useful benchmark problem. We do not focus on the unfolding or parameter estimation steps themselves, but there are many proposals for doing unfolding [10–30] and parameter estimation [31–41] with machine learning. Instead, our focus is on the construction of observables that are engineered to be sensitive to target parameters or to distinguishing models while also insensitive to detector effects. The latter quality ensures that uncertainties arising from the dependence on unfolding 'priors' is small. This is explicitly illustrated using a standard, binned unfolding method in the deep inelastic scattering demonstration.

This paper is organized as follows. Section 2 introduces our approach to observable construction. The datasets used for demonstrating the new method are introduced in Sect. 3. Results with these datasets are presented in Sect. 4. The paper ends with conclusions and outlook in Sect. 5.

2 Methodology

We begin by constructing new observables that are simultaneously sensitive to a parameter while also being minimally sensitive to detector effects. This is accomplished by training neural networks f with the following loss function:

$$L[f] = L_{\text{classic}}[f(z), \mu] + \lambda\, L_{\text{new}}[f(x), f(z)], \tag{1}$$

where $\lambda > 0$ is a hyperparameter that controls how much we regularize the network. We have pairs of inputs (Z, X) where Z represents the particle-level inputs and X represents the detector-level inputs. Capital letters represent random variables while lower-case letter represent realizations of the corresponding random variables. We consider the case X and Z have the same structure, i.e. they are both sets of 4-vectors (so it makes sense to compute $f(z)$ and $f(x)$). This is the standard case where X is a set of energy-flow objects that are meant to correspond to the 4-vectors of particles before being distorted by the detector. Furthermore, we fix the same definition of the observable at particle and detector level.

The first term in Eq. 1, L_{classic}, governs the sensitivity of the observable f to the target parameter μ at particle level. For regression tasks, μ will be a real number, representing e.g. a (dimensionless) mass or coupling. For two-sample tests, $\mu \in \{0, 1\}$, where 0 represents the null hypothesis and 1 represents the alternative hypothesis. A classification setup may also be useful for a regression task, by using two samples at different values of the target parameter. The second term in Eq. 1, L_{new}, governs how sensitive the new observable is to detector effects. It has the property that it is small when $f(x)$ and $f(z)$ are the same and large otherwise. When $\lambda \to \infty$, the observable is completely insensitive to detector effects. This means that any uncertainty associated with removing such effects (including the dependence on unfolding 'priors') is eliminated. The best value of λ will be problem specific and should ideally be chosen based on one or more downstream tasks with the unfolded data.

We introduce the method with a toy model for continuous parameter estimation (Sect. 2.1), which demonstrates the essential ideas in a simplified context. This is followed by a more complete binary classification example using simulated deep inelastic scattering events from the H1 experiment at HERA (Sect. 2.2), where the goal is to be maximally sensitive to distinguishing two datasets.

2.1 Toy example for continuous parameter estimation

The training samples are generated with a uniform distribution for the parameter of interest μ, so each event is specified by (μ_i, z_i, x_i). Then, we parameterize the observable f as a neural network and optimize the following loss function:

$$L[f] = \sum_i (f(z_i) - \mu_i)^2 + \lambda \sum_i (f(x_i) - f(z_i))^2, \tag{2}$$

where the form of both terms is the usual mean squared error loss used in regression tasks. The first term trains the regression to predict the parameter of interest μ while the second term trains the network to make the predictions given detector level features x and particle level features z similar. We use the prediction based on particle-level features z_i in the first term in the loss function. Results for the alternative choice, using the detector-level features x_i are shown in Appendix A.

The loss function in Eq. 2 is similar to the setting of decorrelation, where a classifier is trained to be independent from a given feature [42–60]. One could apply decorrelation techniques in this case to ensure the classifier is not able to distinguish between features at detector level and at particle level. However, this will only ensure that the probability density for f is the same for particle level and detector level. To be well-measured, we need more than statistical similarity between

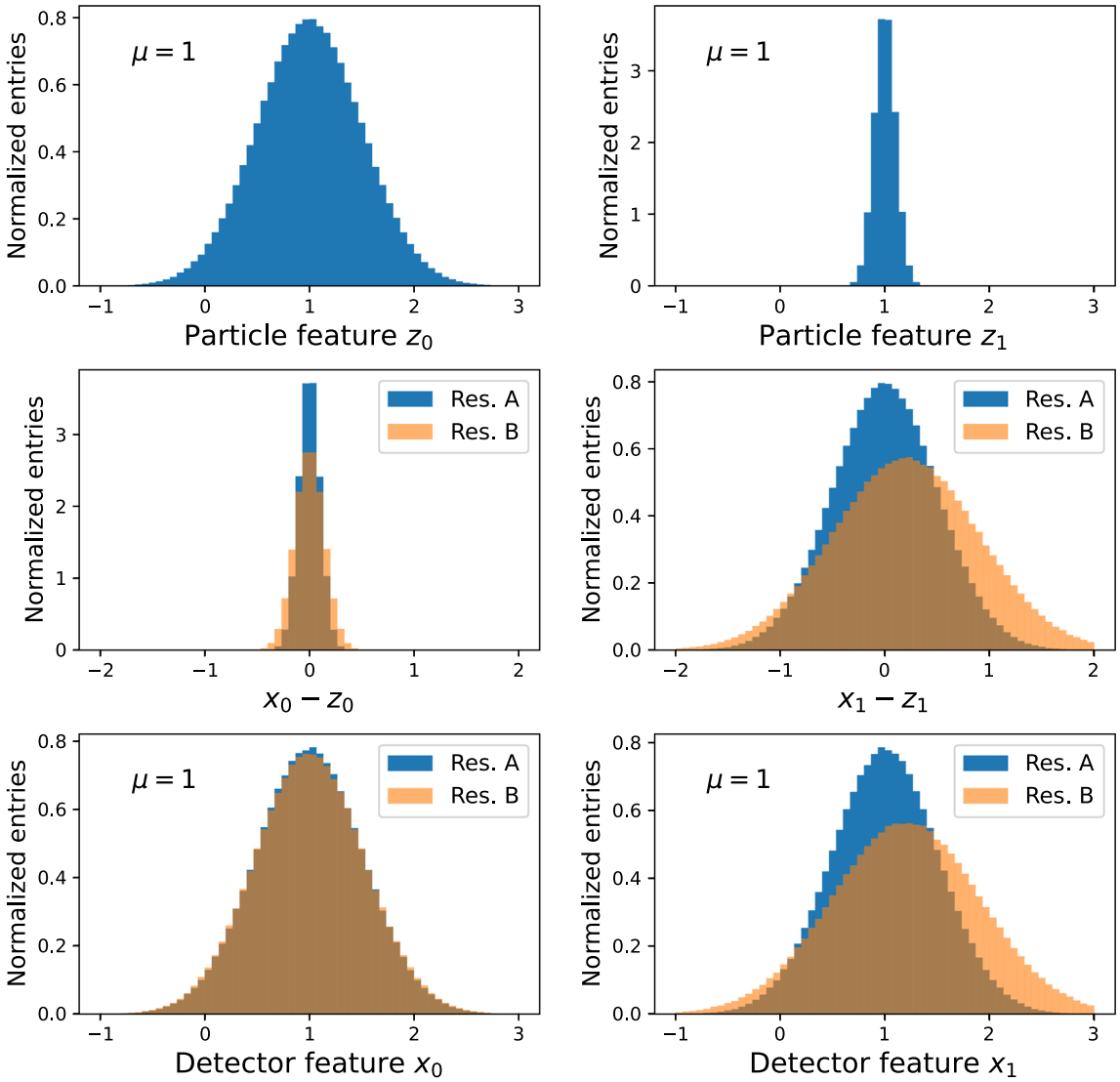

Fig. 1 Input features and resolution model for toy regression example. Two different experimental resolution functions, A and B, are shown

distributions - we need them to be similar event by event. The final term in Eq. 2 is designed for exactly this purpose.

All deep neural networks are implemented in KERAS [61]/ TENSORFLOW [62] and optimized using ADAM [63]. The network models use two hidden layers with 50 nodes per layer and Rectified Linear Unit (ReLU) activation functions for intermediate layers and a linear activation function for the last layer.

Figure 1 illustrates the input features and resolution model for the toy study. Two particle-level features z_0 and z_1 are modeled as normal distributions: $Z_0 \sim \mathcal{N}(\mu, 0.5)$ and $Z_1 \sim \mathcal{N}(\mu, 0.1)$, where feature 1 is significantly more sensitive to the parameter of interest μ. The experimental resolution on the features is given by $X_0 \sim \mathcal{N}(Z_0, 0.1)$ and $X_1 \sim \mathcal{N}(Z_1, 0.5)$ so that feature 0 is well measured, while feature 1 has a relatively poor resolution. For this model, the net experimental sensitivity to μ is the same for both features, but feature 0 is much less sensitive to detector effects. Our proposed method will take this into account in the training of the neural network.

To demonstrate the sensitivity to uncertainties associated with detector effects, we make predictions using f trained with resolution model A on a sample generated with resolution model B, shown in Fig. 1, where the width is increased by a factor of 1.4 for both features and a bias of 0.2 is introduced for the x_1 feature. Figure 2 shows the results as a function of the λ hyperparameter. With $\lambda = 0$, the network relies almost entirely on feature 1, which has better particle-level resolution. As λ increases, more emphasis is placed on feature 0, which has much better detector-level resolution. The resolution of the prediction starts at close to $\sqrt{0.5^2 + 0.1^2}$ for resolution model A, then reaches a minimum close to

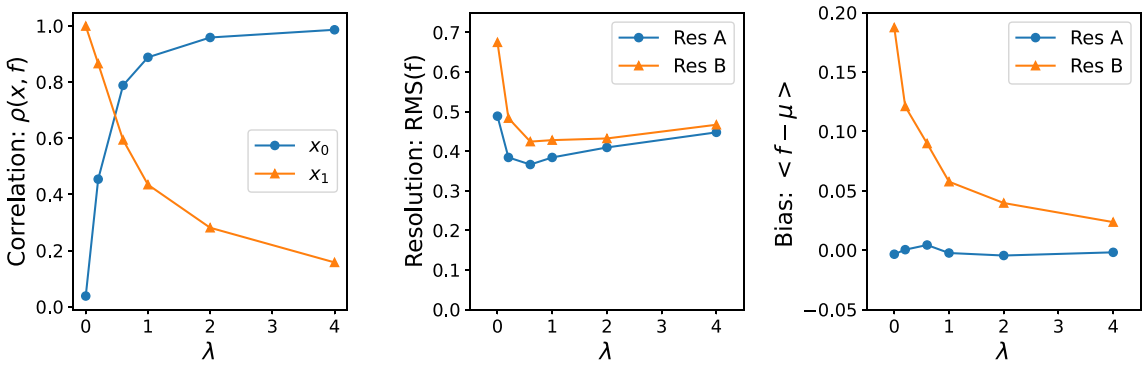

Fig. 2 Results of toy regression example as a function of the λ hyperparameter

$\sqrt{0.5^2 + 0.1^2}/\sqrt{2}$ near $\lambda = 0.5$ where both features have equal weight in the prediction. The bias in the prediction for resolution model B is large for $\lambda = 0$ but falls significantly with increasing λ. This validates the key concept of the proposed method.

2.2 Full example for binary classification

In the second example, we wish to design an observable that will discriminate between two different Monte Carlo parton shower models, while minimizing uncertainties from detector effects. The objective will be to use the distribution of the observable to indicate which model best represents the data. The discrimination test will be successful if the particle-level distribution of the observable for only one of the models is statistically consistent with the unfolded distribution of the observable from data. Ideally, the difference in the shape of the particle-level distributions of the observable for the two models will be large compared to the size of the uncertainty from detector effects in the unfolding.

To design the observable for this task, we train a binary classification neural network to distinguish the two parton shower models, where we minimize detector effects with our additional term in the loss function. The observable is trained to classify events, but this is not a case where each event in the data may be from one of two categories, such as in a neural network trained to discriminate signal from background. All of the events in the data are from the same class. The task is to use the shape of the observable to test which model is more consistent with the data. The neural network is trained to find differences in the input features that allow it to distinguish the two models and this information is reflected in the shape of the neural network output distribution. This makes the shape comparison the appropriate statistical test for the observable.

In the binary case, we have two datasets generated from simulation 1 (sim. 1) and simulation 2 (sim. 2). The loss function for classification is given by

$$L[f] = -\sum_{i\in\text{sim. 1}} \log(f(z_i)) - \sum_{i\in\text{sim. 2}} \log(1 - f(z_i)) + \lambda \times \sum_{i\in\text{sim. 1 \& 2}} (f(x_i) - f(z_i))^2, \quad (3)$$

where the first two terms represent the usual binary cross entropy loss function for classification and the third term represents the usual mean squared error loss term for regression tasks. The notation $i \in$ sim. j means that (z_i, x_i) are drawn from the jth simulation dataset. As in the regression case, the hyperparameter λ must be tuned and controls the trade off between sensitivity to the dataset and sensitivity to detector effects. The network model is the same as in the previous example except that the final layer uses a sigmoid activation function. The binary case is a special case of the previous section where there are only two values of the parameter of interest. It may also be effective to train the binary case for a continuous parameter using two extreme values of the parameter. In this paper, we use high-quality, well-curated datasets from the binary case because of their availability, but it would be interesting to explore the continuous case in the future.

To determine the efficacy of the new observable, we unfold the pseudodata. Unfolding corrects for detector effects by performing regularized matrix inversion on the response matrix. We employ the widely-used TUNFOLD method [64], which is a least-squared-based fit with additional regularization. We use TUNFOLD version 17.9 [64] through the interface included in the ROOT 6.24 [65] distribution. The response matrix is defined from a 2D binning the NN output, given detector level and particle level features. The matrix uses 24 and 12 bins for detector and particle inputs, respectively, which gives reasonable stability and cross-bin correlations in the unfolding results. The ultimate test is to show that the difference between sim. 1 at detector-level unfolded with sim. 2 for the response matrix and the particle level sim. 1 (or vice versa) is smaller than the difference between sim. 1 and sim. 2 at particle level. In other words, this test shows

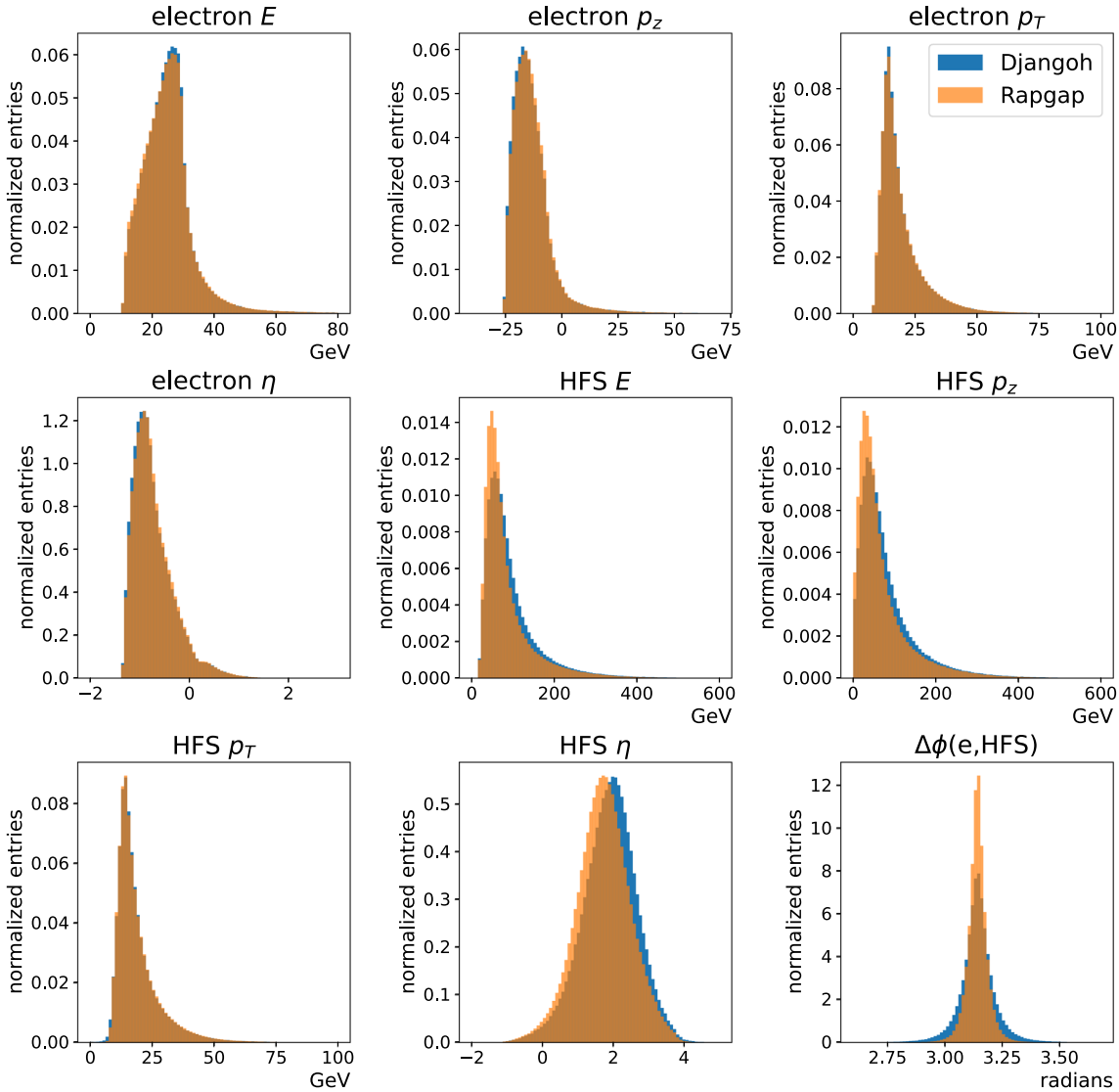

Fig. 3 Particle level distributions of the nine NN input features for the DJANGOH and RAPGAP generators

that the ability to distinguish sim. 1 and sim. 2 significantly exceeds the modeling uncertainty from the unfolding.

3 Datasets

We use deep inelastic scattering events from high-energy electron-proton collisions to demonstrate the performance of the new approach. These simulated data are from the H1 experiment at HERA [66,67] and are used in the same way as Ref. [68]. They are briefly described in the following.

Two parton shower Monte Carlo programs provide the particle-level simulation: RAPGAP 3.1 [69] or DJANGOH 1.4 [70]. The energies of the incoming beams are $E_e = 27.6$ GeV and $E_p = 920$ GeV, for the lepton and proton, respectively, matching the running conditions of HERA II. Radiation from Quantum Electrodynamic processes is simulated by HERACLES routines [71–73] in both cases. The outgoing particles from these two datasets are then fed into a GEANT 3 [74]-based detector simulation.

Following the detector simulation, events are reconstructed with an energy-flow algorithm [75–77] and the scattered electron is reconstructed using the default H1 approach [23,78,79]. Mis-measured backgrounds are suppressed with standard selections [78,79]. This whole process makes use of the modernized H1 computing environment at DESY [80]. Each dataset is comprised of approximately 10 million events.

Figure 3 shows histograms of the nine features used as input for the neural network training. These features include the energy E, longitudinal momentum p_z, transverse momentum p_T, and pseudorapidity η of the scattered elec-

tron and the total Hadronic Final State (HFS), as well as the difference in azimuthal angle between the two $\Delta\phi(e, \text{HFS})$. The HFS is quite sensitive to the η acceptance of the detector. In order to have HFS features that are comparable for the particle and detector definitions, we only use generated final state particles with $|\eta| < 3.8$ in the definition of the particle-level HFS 4-vector. Both simulations provide event weights that must be used for physics analysis. In our study, we do not weight the simulated events in order to maximize the effective statistics of the samples in the neural network training. This has a small effect on the spectra, but has a large impact on the number of effective events available for training. The electron feature distributions agree very well for the two simulations, while there are some visible differences in the HFS features.

4 Results

We now apply the method introduced in Sect. 2.2 to the DIS dataset described in Sect. 3. Figure 4 shows the results of four neural network trainings with different values of λ, which sets the relative weight of the MSE term in the loss function (Eq. 3) that controls the sensitivity to detector effects. With $\lambda = 0$, the classification performance for particle-level inputs is strong, while there are significant disagreements between the particle and detector level neural network outputs. As λ increases, the particle and detector level agreement improves at the cost of weaker classification performance. In what follows, we will use the network trained with $\lambda = 100$. For a parameter estimation task, the entire distribution will be used for inference and therefore excellent event-by-event classification is not required.

Next, we investigate how these spectra are preserved after unfolding. Figure 5 shows the results of unfolding the neural network output, given detector-level features, to give the neural network output distribution for particle-level features. The input distribution for the unfolding is 10^5 events randomly chosen from a histogram of the neural network output for detector-level inputs from the simulation. The unfolding response matrices for the two simulations agree fairly well and are concentrated along the diagonal. The output of the unfolding shows very good agreement with the true distribution of the neural network output given particle-level inputs, demonstrating acceptable closure for the unfolding. The correlations in the unfolding result are mostly between neighboring bins of the distribution.

One of the biggest challenges for the $\lambda = 0$ case is that it is highly sensitive to regions of phase space that are not well-constrained by the detector. As a result, the output of the unfolding is highly dependent on the simulation used in the unfolding (prior). The left side of Fig. 6 shows the model dependence of the unfolding and the ability of the neural network to perform the model classification task. The unfolding model dependence is estimated from a comparison with using the response matrix from the other simulation. We test the model discrimination sensitivity by dividing the unfolded distribution by the particle-level distribution from the other simulation. The degree to which this ratio deviates from unity is a measure of the model discrimination power of the method. For the neural network, the model dependence of the unfolding is small and generally less than 10%. The size of the deviation from unity for the neural network is large compared to the size of the uncertainty, which is dominated by the model dependence, indicating that the neural network can distinguish the two simulations.

Figure 3 shows that some of the HFS variables in the input features may be able to distinguish the two models directly. The right side of Fig. 6 shows the results of running the unfolding procedure using the HFS η distribution for model discrimination, where the model dependence is significantly larger, compared to the neural network output. The shape is distorted, including deviations up to 20%, when the response matrix from the other simulation was used in the unfolding. Since the modeling uncertainty is comparable to or larger than the size of the effect we are trying to probe, such observables are much less useful than the neural network output for the inference task.

We perform a quantitative evaluation of the model discrimination power for this example using a χ^2 computed from the difference between the unfolded distribution and the particle-level distribution for a given model. The uncertainties in the χ^2 are from a combination of the unfolding covariance matrix and a covariance matrix for model dependence uncertainty from the unfolding response matrix. The distribution of the χ^2 from a set of toy Monte Carlo experiments with 2000 events per experiment is close to that from a χ^2 PDF with 12 degrees of freedom when the model for the comparison in the χ^2 is the same as the model used to generate the toy samples (DJANGOH), which validates the unfolding statistical uncertainty in the covariance matrix and the χ^2 calculation. We use this distribution to define a χ^2 threshold for the critical region corresponding to a frequency of 1% for the correct model hypothesis to have a χ^2 greater than the threshold. When the toy samples are drawn from the alternative model (RAPGAP), we find that the frequency for the χ^2 to be above the threshold is 98.6% for the designer neural network observable, but only 63.0% for the HFS η observable. This shows that the designer neural network observable has superior discrimination power.

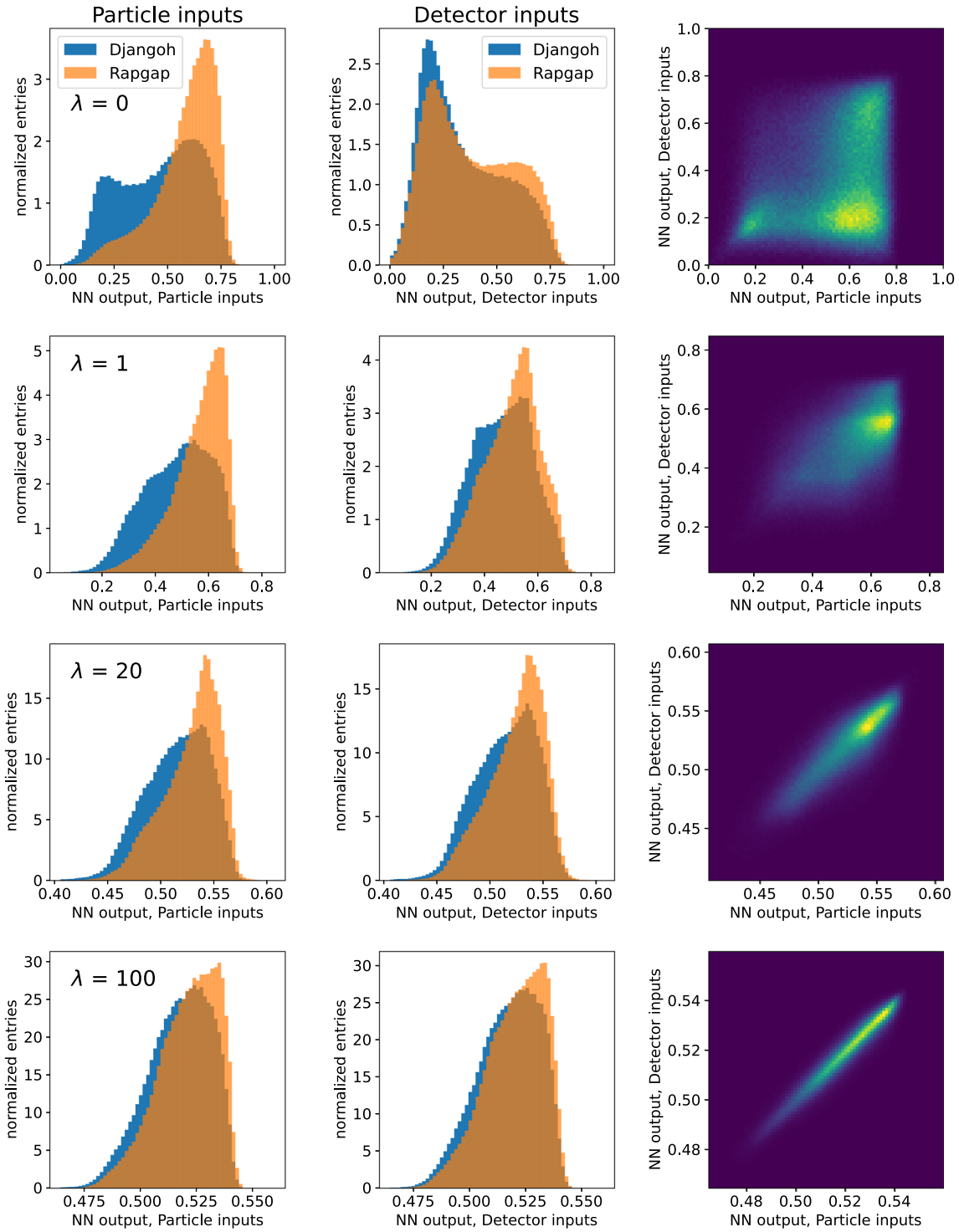

Fig. 4 Neural network output distributions for four values of the λ hyperparameter, which sets the scale for the Detector - Particle disagreement penalty in the loss function. The top row shows the results for $\lambda = 0$, where there is no penalty if the NN predictions for Detector-level input features and Particle-level input features disagree. The bottom three rows show increasing values of λ: 1, 20, and 100

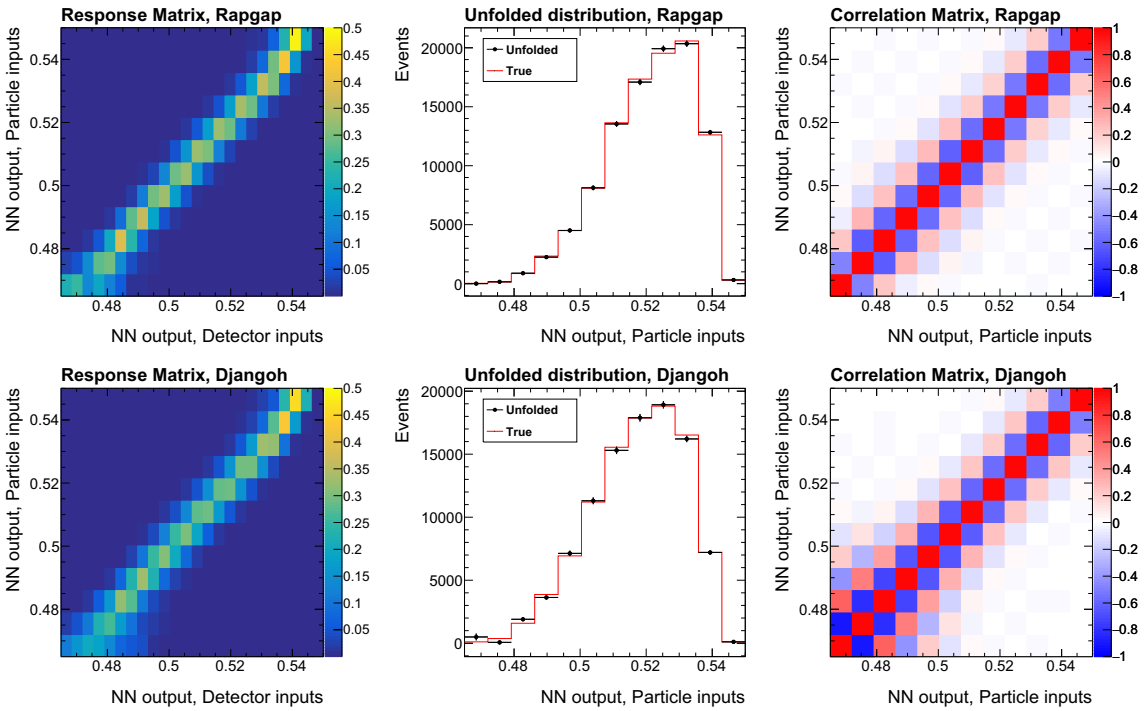

Fig. 5 Results of unfolding the NN output. The top (bottom) row shows the unfolding for the RAPGAP (DJANGOH) generator. The left column shows the response matrix for the unfolding, where the distribution of the NN output given the Detector-based input features (horizontal axis) is normalized to unit area for each bin of the NN output given the Particle-based input features (vertical axis). The center column shows the unfolded distribution compared to the true distribution. The right column shows the matrix of correlation coefficients from the unfolding

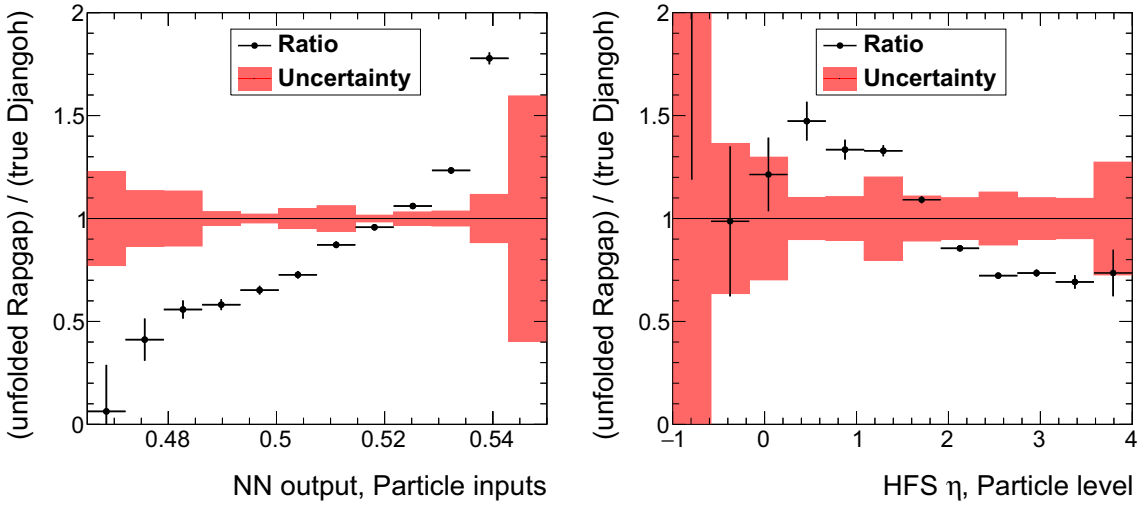

Fig. 6 Model discrimination sensitivity and uncertainty for the neural network output distribution (left) and the hadronic final state η distribution (right). The uncertainty, shown by the shaded red distribution, is the model dependence of the unfolding added in quadrature with the statistical error from the unfolding (error bars on the black points). The black points show the ratio of the unfolded distribution from the RAPGAP simulation divided by the particle-level distribution from the DJANGOH simulation. The significance of the deviation from unity is a measure of the model discrimination sensitivity. Additional distributions and comparisons are given in Appendix B

5 Conclusions and outlook

Unfolded differential cross section measurements are a standard approach to making data available for downstream inference tasks. While some measurements can be used for a variety of tasks, often, there is a single goal that motivates the result. In these cases, we advocate to design observables that are tailored to the physics goal using machine learning. The output of a neural network trained specifically for the downstream task is an observable and its differential cross section likely contains more information than classical observables. We have proposed a new loss function for training the network so that the resulting observable can be well measured. The neural network observable is thus trained using a loss function composed of two parts: one part that regresses the inputs onto a parameter of interest and a second part that penalizes the network for producing different answers at particle level and detector level. A tunable, and problem-specific hyperparameter determines the tradeoff between these two goals. We have demonstrated this approach with both a toy and physics example. For the deep inelastic scattering example, the new approach is shown to be much more sensitive than classical observables while also having a reduced dependence on the starting simulation used in the unfolding. We anticipate that our new approach could be useful for a variety of scientific goals, including measurements of fundamental parameters like the top quark mass and tuning Monte Carlo event generators.

There are a number of ways this approach could be extended in the future. We require that the observable have the same definition at particle level and detector level, while additional information at detector-level like resolutions may be useful to improve precision. A complementary strategy would be to use all the available information to unfold the full phase space [30]. Such techniques may improve the precision by integrating all of the relevant information at detector level, but they may compromise specific sensitivity by being broad and have no direct constraints on measurability. It would be interesting to compare our tailored approach to full phase space methods in the future.

Acknowledgements We thank Miguel Arratia and Daniel Britzger for useful discussions and feedback on the manuscript. Additionally, we thank our colleagues from the H1 Collaboration for allowing us to use the simulated MC event samples. Thanks to DESY-IT and the MPI für Physik for providing some computing infrastructure and supporting the data preservation project of the HERA experiments. B.N. was supported by the Department of Energy, Office of Science under contract number DE-AC02-05CH11231.

Data Availability Statement This manuscript has no associated data. [Author's' comment: Data sharing is not applicable for this work.].

Code Availability Statement This manuscript has associated code/software in a data repository. [Author's comment: The code for this study is available in the following github repository: https://github.com/owen234/designer-obs-paper.]

Funded by SCOAP[3].

Appendix A: Alternative loss function for the regression example

An alternative approach in the regression example is to use the detector-level features instead of the particle-level features in the first term of the loss function. Figure 7 shows the results of using Eq. A.1 instead of Eq. 2 in the training.

$$L[f] = \sum_i (f(x_i) - \mu_i)^2 + \lambda \sum_i (f(x_i) - f(z_i))^2, \quad \text{(A.1)}$$

With $\lambda = 0$, which corresponds to the usual approach for this type of regression task, the correlation between each of the detector level features x_0 and x_1 and the network prediction f is the same, reflecting the fact that x_0 and x_1 have the same sensitivity to μ. As λ increases, more emphasis is placed on feature x_0, which is well measured. The resolution of the regression, given by the RMS of f, starts at the expected value of about $\sqrt{0.5^2 + 0.1^2}/\sqrt{2}$ for resolution model A and $\lambda = 0$ and increases with λ as the network relies more on x_0 for the prediction.

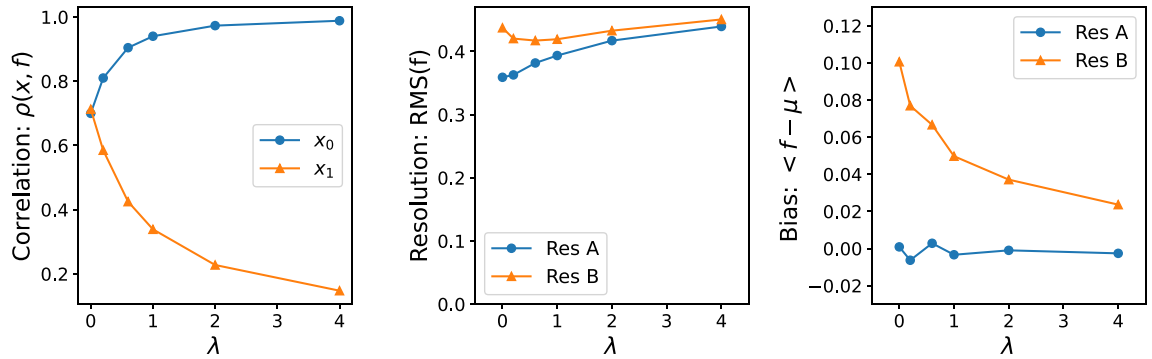

Fig. 7 Results of toy regression example as a function of the λ hyperparameter, using Eq. A.1 for the loss function in the training

Appendix B: Additional distributions for the model dependence and discrimination tests

Figures 8 and 9 show the model dependence of the unfolding and model discrimination sensitivity for the neural network and the HFS η distributions, respectively. The top row of each figure shows the results of the closure tests compared with a measure of the unfolding model dependence, where we perform the unfolding with the response matrix from the other simulation. The unfolded distributions have been normalized

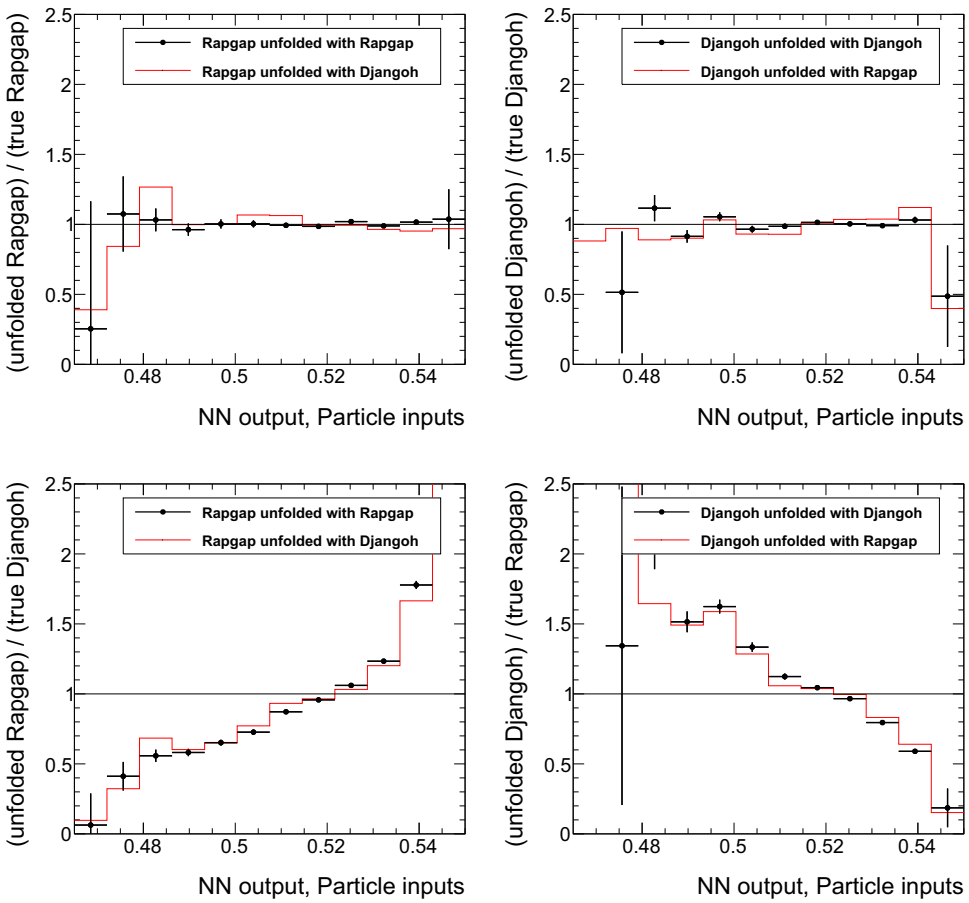

Fig. 8 Results of testing the model dependence of the unfolding the **neural network output** distribution by varying the response matrix used in the unfolding. The top row normalizes the unfolded distribution with the true distribution from the same simulation, testing the unfolding closure (points) and unfolding model dependence (red histogram). The bottom row normalizes the unfolded distribution with the true distribution from the other simulation, showing the ability to distinguish the models

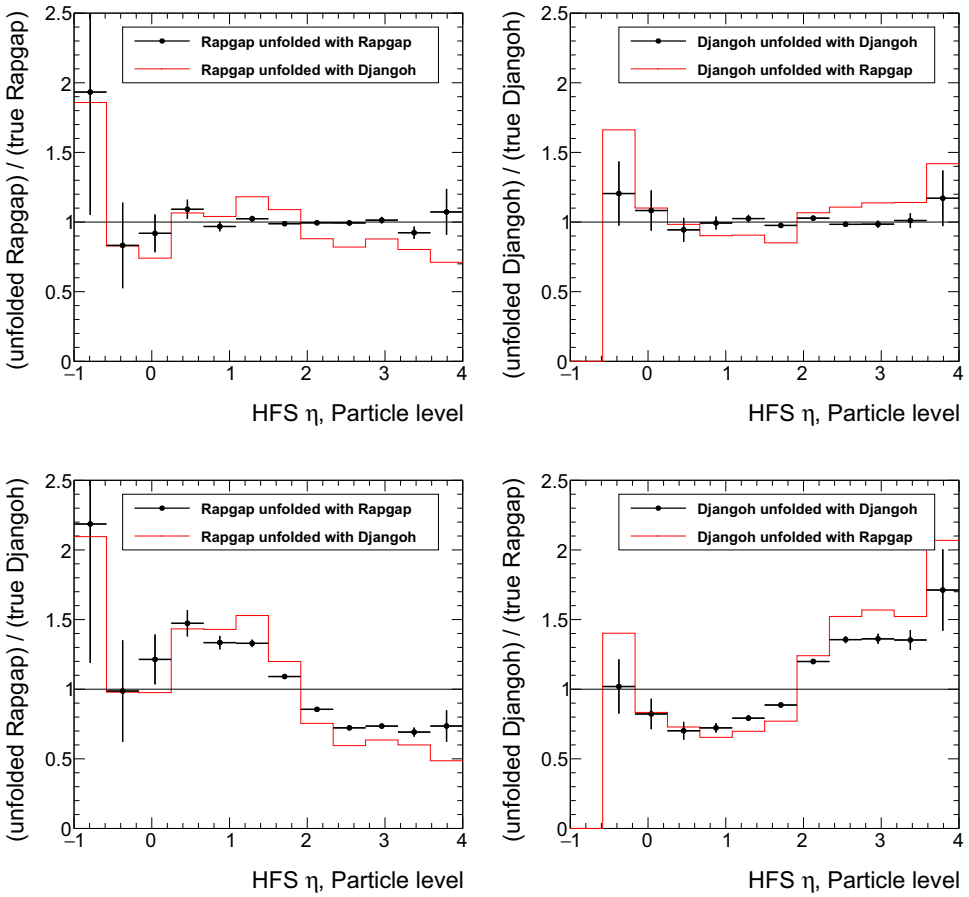

Fig. 9 Results of testing the model dependence of the unfolding the **HFS** η distribution by varying the response matrix used in the unfolding. The top row normalizes the unfolded distribution with the true distribution from the same simulation, testing the unfolding closure (points) and unfolding model dependence (red histogram). The bottom row normalizes the unfolded distribution with the true distribution from the other simulation, showing the ability to distinguish the models

using the true distribution from the same simulation, giving an expected flat distribution consistent with 1. The bottom row shows the unfolded distributions instead normalized by the true distribution from the other simulation. The degree to which the ratio deviates from unity is a measure of the model discrimination power of the network. Figure 6 in the main text uses the lower-left distribution from Figs. 8 and 9 to display the results.

The unfolding shows good closure for both the neural network and the HFS η distributions in both simulations. The model dependence of the unfolding for the HFS η distribution is significantly larger than for the neural network distribution. The size of the deviation from unity in the discrimination test for the neural network is large compared to the size of the unfolding model dependence.

References

1. CMS Collaboration, A. Tumasyan et al., Measurement of the differential $\mathrm{t\bar{t}}$ production cross section as a function of the jet mass and extraction of the top quark mass in hadronic decays of boosted top quarks. Eur. Phys. J. C **83**(7), 560 (2023). https://doi.org/10.1140/epjc/s10052-023-11587-8. arXiv:2211.01456
2. H1 Collaboration, V. Andreev, et al., Determination of the strong coupling constant $\alpha_s(m_Z)$ in next-to-next-to-leading order QCD using H1 jet cross section measurements. Eur. Phys. J. C 77 (11) (2017) 791 [Erratum: Eur.Phys.J.C 81, 738 (2021)]. https://doi.org/10.1140/epjc/s10052-017-5314-7. arXiv:1709.07251
3. ZEUS Collaboration, Z. Collaboration, Measurement of jet production in deep inelastic scattering and NNLO determination of the strong coupling at ZEUS (9 2023). arXiv:2309.02889
4. H1, ZEUS Collaboration, H. Abramowicz, et al., Combination of measurements of inclusive deep inelastic $e^{\pm}p$ scattering cross sections and QCD analysis of HERA data. Eur. Phys. J. C 75 (2015). https://doi.org/10.1140/epjc/s10052-015-3710-4. arXiv:1506.06042

5. L.A. Harland-Lang, A.D. Martin, P. Motylinski, R.S. Thorne, The impact of the final HERA combined data on PDFs obtained from a global fit. Eur. Phys. J. C **76**(4), 186 (2016). https://doi.org/10.1140/epjc/s10052-016-4020-1. arXiv:1601.03413
6. T.-J. Hou et al., New CTEQ global analysis of quantum chromodynamics with high-precision data from the LHC. Phys. Rev. D **103**(1), 014013 (2021). https://doi.org/10.1103/PhysRevD.103.014013. arXiv:1912.10053
7. NNPDF Collaboration, R.D. Ball, et al., The path to proton structure at 1% accuracy. Eur. Phys. J. C 82(5), 428. (2022)https://doi.org/10.1140/epjc/s10052-022-10328-7. arXiv:2109.02653
8. J.M. Campbell, et al., Event Generators for High-Energy Physics Experiments, in: Snowmass 2021. (2022). arXiv:2203.11110
9. M. Arratia, D. Britzger, O. Long, B. Nachman, Optimizing observables with machine learning for better unfolding. JINST **17**(07), P07009 (2022). https://doi.org/10.1088/1748-0221/17/07/P07009. arXiv:2203.16722
10. K. Datta, D. Kar, D. Roy, Unfolding with generative adversarial networks (2018). arXiv:1806.00433
11. M. Bunse, N. Piatkowski, T. Ruhe, W. Rhode, K. Morik, Unification of deconvolution algorithms for Cherenkov astronomy, in: *5th International Conference on Data Science and Advanced Analytics (DSAA)* (IEEE, 2018), pp. 21–30
12. T. Ruhe, T. Voigt, M. Wornowizki, M. Börner, W. Rhode, K. Morik, Mining for spectra—the Dortmund spectrum estimation algorithm. Astronomical Society of the Pacific Conference Series Vol. 521, p 394 (2019) http://aspbooks.org/custom/publications/paper/521-0394.html
13. A. Andreassen, P.T. Komiske, E.M. Metodiev, B. Nachman, J. Thaler, OmniFold: a method to simultaneously unfold all observables. Phys. Rev. Lett. **124**, 182001 (2020). https://doi.org/10.1103/PhysRevLett.124.182001. arXiv:1911.09107,
14. M. Bellagente, A. Butter, G. Kasieczka, T. Plehn, R. Winterhalder, How to GAN away detector effects (2019). https://doi.org/10.21468/SciPostPhys.8.4.070. arXiv:1912.00477
15. M. Bellagente, A. Butter, G. Kasieczka, T. Plehn, A. Rousselot, R. Winterhalder, Invertible networks or partons to detector and back again. (2020). https://doi.org/10.21468/SciPostPhys.9.5.074. arXiv:2006.06685
16. M. Vandegar, M. Kagan, A. Wehenkel, G. Louppe, Neural empirical bayes: source distribution estimation and its applications to simulation-based inference, in: *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics, Vol. 130 of Proceedings of Machine Learning Research, PMLR*, ed. by A. Banerjee, K. Fukumizu (2021), pp. 2107–2115. arXiv:2011.05836
17. A. Andreassen, P.T. Komiske, E.M. Metodiev, B. Nachman, A. Suresh, J. Thaler, Scaffolding simulations with deep learning for high-dimensional deconvolution, in: *9th International Conference on Learning Representations* (2021). arXiv:2105.04448
18. J.N. Howard, S. Mandt, D. Whiteson, Y. Yang, Foundations of a fast, data-driven, machine-learned simulator. (2021). arXiv:2101.08944
19. M. Backes, A. Butter, M. Dunford, B. Malaescu, An unfolding method based on conditional invertible neural networks (cINN) using iterative training (12 2022). arXiv:2212.08674
20. J. Chan, B. Nachman, Unbinned profiled unfolding. Phys. Rev. D **108**(1), 016002 (2023). https://doi.org/10.1103/PhysRevD.108.016002. arXiv:2302.05390
21. A. Shmakov, K. Greif, M. Fenton, A. Ghosh, P. Baldi, D. Whiteson, End-To-end latent variational diffusion models for inverse problems in high energy physics (5 2023). arXiv:2305.10399
22. T. Alghamdi et al., Toward a generative modeling analysis of CLAS exclusive 2π photoproduction (7 2023). arXiv:2307.04450
23. H1 Collaboration, V. Andreev et al., Measurement of lepton-jet correlation in deep-inelastic scattering with the H1 detector using machine learning for unfolding (Aug 2021). arXiv:2108.12376
24. H1 Collaboration, Machine learning-assisted measurement of multi-differential lepton-jet correlations in deep-inelastic scattering with the H1 detector, H1prelim-22-031 (2022). https://www-h1.desy.de/h1/www/publications/htmlsplit/H1prelim-22-031.long.html
25. H1 Collaboration, V. Andreev et al., Unbinned deep learning jet substructure measurement in high Q^2 ep collisions at HERA (3 2023). arXiv:2303.13620
26. H1 Collaboration, Measurement of lepton-jet correlations in high Q^2 neutral-current DIS with the H1 detector at HERA, H1prelim-21-031 (2021). https://www-h1.desy.de/h1/www/publications/htmlsplit/H1prelim-21-031.long.html
27. LHCb Collaboration, Multidifferential study of identified charged hadron distributions in Z-tagged jets in proton–proton collisions at $\sqrt{s}$ =13 TeV (8 2022). arXiv:2208.11691
28. P.T. Komiske, S. Kryhin, J. Thaler, Disentangling quarks and gluons in CMS open data. Phys. Rev. D **106**(9), 094021 (2022). https://doi.org/10.1103/PhysRevD.106.094021. arXiv:2205.04459
29. STAR Collaboration, Y. Song, Measurement of CollinearDrop jet mass and its correlation with SoftDrop groomed jet substructure observables in $\sqrt{s}$ = 200 GeV pp collisions by STAR (7 2023). arXiv:2307.07718
30. M. Arratia et al., Publishing unbinned differential cross section results, JINST 17 (01) (2022) P01024. https://doi.org/10.1088/1748-0221/17/01/P01024. arXiv:2109.13243
31. J. Brehmer, K. Cranmer, G. Louppe, J. Pavez, Constraining effective field theories with machine learning. Phys. Rev. Lett. **121**(11), 111801 (2018). https://doi.org/10.1103/PhysRevLett.121.111801. arXiv:1805.00013
32. J. Brehmer, K. Cranmer, G. Louppe, J. Pavez, A guide to constraining effective field theories with machine learning. Phys. Rev. D **98**(5), 052004 (2018). https://doi.org/10.1103/PhysRevD.98.052004. arXiv:1805.00020
33. J. Brehmer, G. Louppe, J. Pavez, K. Cranmer, Mining gold from implicit models to improve likelihood-free inference. Proc. Natl. Acad. Sci. **117**(10), 5242–5249 (2020). https://doi.org/10.1073/pnas.1915980117. arXiv:1805.12244
34. P. De Castro, T. Dorigo, INFERNO: inference-aware neural optimisation. Comput. Phys. Commun. **244**, 170–179 (2019). https://doi.org/10.1016/j.cpc.2019.06.007. arXiv:1806.04743
35. A. Elwood, D. Krücker, Direct optimisation of the discovery significance when training neural networks to search for new physics in particle colliders (6 2018). arXiv:1806.00322
36. A. Andreassen, B. Nachman, Neural networks for full phase-space reweighting and parameter tuning. Phys. Rev. D **101**(9), 091901 (2020). https://doi.org/10.1103/PhysRevD.101.091901. arXiv:1907.08209
37. J. Brehmer, F. Kling, I. Espejo, K. Cranmer, MadMiner: machine learning-based inference for particle physics. Comput. Softw. Big Sci. **4**(1), 3 (2020). https://doi.org/10.1007/s41781-020-0035-2. arXiv:1907.10621
38. S. Wunsch, S. Jörger, R. Wolf, G. Quast, Optimal statistical inference in the presence of systematic uncertainties using neural network optimization based on binned Poisson likelihoods with nuisance parameters. Comput. Softw. Big Sci. **5**(1), 4 (2021). https://doi.org/10.1007/s41781-020-00049-5. arXiv:2003.07186
39. A. Ghosh, B. Nachman, D. Whiteson, Uncertainty-aware machine learning for high energy physics. Phys. Rev. D **104**(5), 056026 (2021). https://doi.org/10.1103/PhysRevD.104.056026. arXiv:2105.08742
40. R. Gomez Ambrosio, J. ter Hoeve, M. Madigan, J. Rojo, V. Sanz, Unbinned multivariate observables for global SMEFT analyses

from machine learning. JHEP **03**, 033 (2023). https://doi.org/10.1007/JHEP03(2023)033. arXiv:2211.02058
41. N. Simpson, L. Heinrich, Neos: end-to-end-optimised summary statistics for high energy physics. J. Phys. Conf. Ser. **2438**(1), 012105 (2023). https://doi.org/10.1088/1742-6596/2438/1/012105. arXiv:2203.05570
42. A. Blance, M. Spannowsky, P. Waite, Adversarially-trained autoencoders for robust unsupervised new physics searches. JHEP **10**, 047 (2019). https://doi.org/10.1007/JHEP10(2019)047. arXiv:1905.10384
43. C. Englert, P. Galler, P. Harris, M. Spannowsky, Machine learning uncertainties with adversarial neural networks. Eur. Phys. J. C **79**(1), 4 (2019). https://doi.org/10.1140/epjc/s10052-018-6511-8. arXiv:1807.08763
44. G. Louppe, M. Kagan, K. Cranmer, Learning to pivot with adversarial networks, in: *Advances in Neural Information Processing Systems*, vol. 30, ed. by I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Curran Associates, Inc., 2017). arXiv:1611.01046. https://papers.nips.cc/paper/2017/hash/48ab2f9b45957ab574cf005eb8a76760-Abstract.html
45. J. Dolen, P. Harris, S. Marzani, S. Rappoccio, N. Tran, Thinking outside the ROCs: designing decorrelated taggers (DDT) for jet substructure. JHEP **05**, 156 (2016). https://doi.org/10.1007/JHEP05(2016)156. arXiv:1603.00027
46. I. Moult, B. Nachman, D. Neill, Convolved substructure: analytically decorrelating jet substructure observables (2017). arXiv:1710.06859
47. J. Stevens, M. Williams, uBoost: a boosting method for producing uniform selection efficiencies from multivariate classifiers. JINST **8**, P12013 (2013). https://doi.org/10.1088/1748-0221/8/12/P12013. arXiv:1305.7248
48. C. Shimmin, P. Sadowski, P. Baldi, E. Weik, D. Whiteson, E. Goul, A. Sogaard, Decorrelated jet substructure tagging using adversarial neural networks. Phys. Rev. D **96**(7), 074034 (2017). https://doi.org/10.1103/PhysRevD.96.074034. arXiv:1703.03507
49. L. Bradshaw, R. K. Mishra, A. Mitridate, B. Ostdiek, Mass agnostic jet taggers (2019). https://doi.org/10.21468/SciPostPhys.8.1.011. arXiv:1908.08959
50. Performance of mass-decorrelated jet substructure observables for hadronic two-body decay tagging in ATLAS, Technical report ATL-PHYS-PUB-2018-014, CERN, Geneva (2018). https://cds.cern.ch/record/2630973
51. G. Kasieczka, D. Shih, DisCo Fever: robust networks through distance correlation. (2020). https://doi.org/10.1103/PhysRevLett.125.122001. arXiv:2001.05310
52. S. Wunsch, S. Jórger, R. Wolf, G. Quast, Reducing the dependence of the neural network function to systematic uncertainties in the input space. (2019). https://doi.org/10.1007/s41781-020-00037-9. arXiv:1907.11674
53. A. Rogozhnikov, A. Bukva, V. Gligorov, A. Ustyuzhanin, M. Williams, New approaches for boosting to uniformity. JINST 10(03), T03002 (2015). https://doi.org/10.1088/1748-0221/10/03/T03002. arXiv:1410.4140
54. C. Collaboration, A deep neural network to search for new long-lived particles decaying to jets, Mach. Learn. Sci. Technol. (2020). https://doi.org/10.1088/2632-2153/ab9023. arXiv:1912.12238
55. G. Kasieczka, B. Nachman, M. D. Schwartz, D. Shih, ABCDisCo: automating the ABCD method with machine learning (2020). https://doi.org/10.1103/PhysRevD.103.035021. arXiv:2007.14400
56. O. Kitouni, B. Nachman, C. Weisser, M. Williams, Enhancing searches for resonances with machine learning and moment decomposition (2020). arXiv:2010.09745
57. V. Estrade, C. Germain, I. Guyon, D. Rousseau, Systematic aware learning—a case study in high energy physics. EPJ Web Conf. **214**, 06024 (2019). https://doi.org/10.1051/epjconf/201921406024
58. J.A. Aguilar-Saavedra, J.H. Collins, R.K. Mishra, A generic anti-QCD jet tagger. JHEP **11**, 163 (2017). https://doi.org/10.1007/JHEP11(2017)163. arXiv:1709.01087
59. J.A. Aguilar-Saavedra, F.R. Joaquim, J.F. Seabra, Mass unspecific supervised tagging (MUST) for boosted jets. (2020). https://doi.org/10.1007/JHEP03(2021)012. arXiv:2008.12792
60. J.A. Aguilar-Saavedra, E. Arganda, F.R. Joaquim, R.M. Sandá Seoane, J.F. Seabra, Gradient boosting MUST taggers for highly-boosted jets (5 2023). arXiv:2305.04957
61. F. Chollet, Keras. (2015). https://github.com/fchollet/keras
62. 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*, vol. 16 (2016), pp. 265–283
63. D. Kingma, J. Ba, Adam: a method for stochastic optimization. (2014). arXiv:1412.6980
64. S. Schmitt, TUnfold: an algorithm for correcting migration effects in high energy physics. JINST **7**, T10003 (2012). https://doi.org/10.1088/1748-0221/7/10/T10003. arXiv:1205.6201
65. R. Brun, F. Rademakers, ROOT: an object oriented data analysis framework. Nucl. Instrum. Methods A **389**, 81–86 (1997). https://doi.org/10.1016/S0168-9002(97)00048-X
66. H1 Collaboration, I. Abt et al., The tracking, calorimeter and muon detectors of the H1 experiment at HERA. Nucl. Instrum. Methods A **386**, 348–396 (1997). https://doi.org/10.1016/S0168-9002(96)00894-7
67. H1 Collaboration, I. Abt et al., The H1 detector at HERA. Nucl. Instrum. Methods A **386**, 310–347 (1997). https://doi.org/10.1016/S0168-9002(96)00893-5
68. M. Arratia, D. Britzger, O. Long, B. Nachman, Reconstructing the kinematics of deep inelastic scattering with deep learning. Nucl. Instrum. Methods A **1025**, 166164 (2022). https://doi.org/10.1016/j.nima.2021.166164. arXiv:2110.05505
69. H. Jung, Hard diffractive scattering in high-energy e p collisions and the Monte Carlo generator RAPGAP. Comput. Phys. Commun. **86**, 147–161 (1995). https://doi.org/10.1016/0010-4655(94)00150-Z
70. K. Charchula, G.A. Schuler, H. Spiesberger, Combined QED and QCD radiative effects in deep inelastic lepton–proton scattering: the Monte Carlo generator DJANGO6. Comput. Phys. Commun. **81**, 381–402 (1994). https://doi.org/10.1016/0010-4655(94)90086-8
71. H. Spiesberger et al., Radiative corrections at HERA (1992) 798–839. https://cds.cern.ch/record/237380
72. A. Kwiatkowski, H. Spiesberger, H.J. Mohring, Characteristics of radiative events in deep inelastic ep scattering at HERA. Z. Phys. C **50**, 165–178 (1991). https://doi.org/10.1007/BF01558572
73. A. Kwiatkowski, H. Spiesberger, H.J. Mohring, Heracles: an event generator for *ep* interactions at HERA energies including radiative processes: version 1.0. Comput. Phys. Commun. **69**, 155–172 (1992). https://doi.org/10.1016/0010-4655(92)90136-M
74. S. Agostinelli et al., Geant4—a simulation toolkit. Nucl. Instrum. Methods Phys. Res. Sect. A Accelerators Spectrom. Detectors Assoc. Equip. **506**(3), 250–303 (2003). https://doi.org/10.1016/S0168-9002(03)01368-8
75. M. Peez, Search for deviations from the standard model in high transverse energy processes at the electron proton collider HERA. (Thesis, Univ. Lyon), Ph.D. thesis (2003)
76. S. Hellwig, Untersuchung der D^* - π slow Double Tagging Methode in Charmanalysen, Diploma thesis, Univ. Hamburg (2004)
77. B. Portheault, First measurement of charged and neutral current cross sections with the polarized positron beam at HERA II and QCD-electroweak analyses. (Thesis, Univ. Paris XI), Ph.D. thesis (2005)
78. H1 Collaboration, F.D. Aaron et al., Inclusive deep inelastic scattering at high Q^2 with longitudinally polarised lepton beams at HERA.

JHEP **09**, 061. (2012) https://doi.org/10.1007/JHEP09(2012)061. arXiv:1206.7007

79. H1 Collaboration, V. Andreev et al., Measurement of multijet production in ep collisions at high Q^2 and determination of the strong coupling α_s. Eur. Phys. J. C **75**(2), 65 (2015). https://doi.org/10.1140/epjc/s10052-014-3223-6. arXiv:1406.4709
80. D. Britzger, S. Levonian, S. Schmitt, D. South, Preservation through modernisation: the software of the H1 experiment at HERA. EPJ Web Conf. **251**, 03004 (2021). https://doi.org/10.1051/epjconf/202125103004. arXiv:2106.11058