Unbinned Profiled Unfolding

Jay Chan [1 2] **Benjamin Nachman** [2 3]

Abstract

Unfolding is an important procedure in particle physics experiments which corrects for detector effects and provides differential cross section measurements that can be used for a number of downstream tasks, such as extracting fundamental physics parameters. Traditionally, unfolding is done by discretizing the target phase space into a finite number of bins and is limited in the number of unfolded variables. Recently, there have been a number of proposals to perform unbinned unfolding with machine learning. However, none of these methods (like most unfolding methods) allow for simultaneously constraining (profiling) nuisance parameters. We propose a new machine learning-based unfolding method that results in an unbinned differential cross section and can profile nuisance parameters. The machine learning loss function is the full likelihood function, based on binned inputs at detector-level. We demonstrate the method and show the impact on a simulated Higgs boson cross section measurement.

1. Introduction

One of the most common analysis goals in particle and nuclear physics is the measurement of differential cross sections. These quantities encode the rate at which a particular process occurs as a function of certain observables of interest. From measured cross sections, a number of downstream inference tasks can be performed, including the estimation of fundamental parameters, tuning simulations, and searching for physics beyond the Standard Model. The key challenge of cross section measurements is correcting the data for detector distortions, a process called deconvolution or *unfolding*. See Refs. (Cowan, 2002; Blobel, 2011; 2013; Balasubramanian et al., 2019) for recent reviews on unfolding and Refs. (D'Agostini, 1995; Hocker & Kartvelishvili, 1996; Schmitt, 2012) for the most widely-used unfolding algorithms.

Until recently, all cross section measurements were performed with histograms. In particular, the target spectra and experimental observations were binned and the unfolding problem is recast in the language of linear algebra. That is, one would like to determine the signal strength, defined as the ratio of the observed signal yield to the theoretical prediction, for each bin based on the measurements from experimental observations. This approach comes with the limitation that the binning must be determined beforehand. This makes it difficult to compare measurements with different binning. Furthermore, the optimal binning depends on the downstream inference task.

Modern machine learning (ML) has enabled the creation of unfolding methods that can process unbinned data (Arratia et al., 2022). Deep generative models such as Generative Adversarial Networks (GAN) (Goodfellow et al., 2014; Datta et al., 2018; Bellagente et al., 2019) and Variational Autoencoders (VAE) (Kingma & Welling, 2014; Howard et al., 2021) produce implicit models that represents the probability density of the unfolded result and allow to sample from the probability density. Methods based on Normalizing Flows (NF) (Rezende & Mohamed, 2015; Bellagente et al., 2020; Vandegar et al., 2021; Backes et al., 2022) allow for both sampling and density estimation. In contrast, the classifier-based method OmniFold Refs. (Andreassen et al., 2020; 2021) iteratively reweights a simulated dataset. A summary of machine learning-based unfolding methods can be found in Ref. (Arratia et al., 2022) and recent applications of these techniques (in particular, of OmniFold) to experimental data are presented in Refs. (H1 Collaboration, 2022a;b;c; LHCb Collaboration, 2022). While powerful, none of these approaches can simultaneously estimate cross sections and fit (nuisance) parameters. This can be a significant shortcoming when the phase space region being probed has non-trivial constraining power for systematic uncertainties.

Unfolding methods that can also profile have been proposed. One possibility is to treat the cross section in each region

[1]Department of Physics, University of Wisconsin-Madison, Madison, WI 53706, 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. Correspondence to: Jay Chan <jaychan@lbl.gov>.

Accepted after peer-review at the 1st workshop on Synergy of Scientific and Machine Learning Modeling, SynS & ML ICML, Honolulu, Hawaii, USA. July, 2023.

of particle-level phase space (i.e. in a histogram bin) as a free parameter and then perform a likelihood fit as for any set of parameters of interest and nuisance parameters. For example, this is the setup of the the Simplified Template Cross Section (STXS) (e.g. Refs. (de Florian et al., 2016; Andersen et al., 2016; Berger et al., 2019; Amoroso et al., 2020)) measurements for Higgs boson kinematic properties. Another possibility is Fully Bayesian Unfolding (FBU) (Choudalakis, 2012), which samples from the posterior probability over the cross section in each bin of the particle-level phase space and over the nuisance parameters. All of these methods require binning.

In this paper, we propose a new machine learning-based unfolding method that is both unbinned at particle level and can profile, referred to as Unbinned Profiled Unfolding (UPU). UPU reuses all the standard techniques used in binned maximum likelihood unfolding and combines them with ML methods that allow for unbinned unfolding. Specifically, we use the binned maximum likelihood at detector level as the metric to optimize the unfolding, while the unfolding takes unbinned particle-level simulations as inputs.

2. Unbinned Profiled Unfolding

2.1. Machine Learning Approach

UPU generalizes binned maximum likelihood unfolding to the unbinned case. Data at detector-level are still treated as binned in order to know the likelihood (Poisson) while the corresopnding particle-level (pre-detector) data are unbinned. Bin-free results are achieved by learning a function to reweight samples from an initial simulated sample.

For particle-level features T and detector-level features R, the main goal is to train the likelihood ratio estimator $w_0\,(T)$, which reweights the simulated particle-level spectrum. In the absence of profiling, this corresponds to the following loss function:

$$L = \prod_{i=1}^{n_{\text{bins}}} \Pr\left(n_i \middle| \sum_{j=1}^{n_{\text{MC}}} w_0(T_j)\mathbb{I}_i(R_j)\right) , \tag{1}$$

where n_i is the number of observed events in bin i, n_{MC} is the number of simulated events, and $\mathbb{I}_i(\cdot)$ is the indicator function that is one when $\cdot$ is in bin i and zero otherwise. When w_0 is parameterized as a neural network (see Section 2.2), then the logarithm of Equation (1) is used for training:

$$\log L = \tag{2}$$
$$\sum_{i=1}^{n_{\text{bins}}} \left[n_i \log\left(\sum_{j=1}^{n_{\text{MC}}} w_0(T_j)\mathbb{I}_i(R_j)\right) - \sum_{i=1}^{n_{\text{MC}}} w_0(T_j)\mathbb{I}_i(R_j)\right] ,$$

where we have dropped constants that do not affect the optimization. Experimental nuisance parameters modify the predicted counts in a particular bin given the particle-level counts. We account for these effects with a second reweighting function:

$$w_1(R|T,\theta) = \frac{p_\theta(R|T)}{p_{\theta_0}(R|T)} , \tag{3}$$

where $p_\theta(R|T)$ is the conditional probability density of R given T with nuisance parameters θ. Importantly, w_1 does not modify the target particle level distribution. Incorporating w_1 into the log likelihood results in the full loss function:

$$\log L = \sum_{i=1}^{n_{\text{bins}}} \left[n_i \log\left(\sum_{j=1}^{n_{\text{MC}}} w_0(T_j) w_1(R_j|T_j,\theta)\mathbb{I}_i(R_j)\right) - \sum_{j=1}^{n_{\text{MC}}} w_0(T_j) w_1(R_j|T_j,\theta)\mathbb{I}_i(R_j)\right] + \log p_0(\theta) . \tag{4}$$

Since w_1 does not depend on the particle-level spectrum, it can be estimated prior to the final fit and only the parameters of w_0 and the value(s) of θ are allowed to float when optimizing Equation (4).

2.2. Machine Learning Implementation

In our subsequent case studies, the reweighting functions w_0 and w_1 are parametrized with neural networks. The w_0 function is only constrained to be non-negative and so we choose it to be the exponential of a neural network.

The pre-training of w_1 requires neural conditional reweighting (Nachman & Thaler, 2022), as a likelihood ratio in R conditioned on T and parameterized in θ. While there are multiple ways of approximating conditional likelihood ratios, the one we found to be the most stable for the examples we have studied for UPU is the product approach:

$$w_1(R|T,\theta) = \left(\frac{p_\theta(R,T)}{p_{\theta_0}(R,T)}\right)\left(\frac{p_{\theta_0}(T)}{p_\theta(T)}\right) , \tag{5}$$

where the two terms on the righthand side are separately estimated and then their product is w_1. For a single feature T,

a likelihood ratio between samples drawn from a probability density p and samples drawn from a probability density q is estimated using the fact that machine learning-classifiers approximate monotonic transformations of likelihood ratios (see e.g. Ref. (Hastie et al., 2001; Sugiyama et al., 2012)). In particular, we use the standard binary cross entropy loss function

$$L_{\text{BCE}}[f] = -\sum_{Y\sim p} \log(f(Y)) - \sum_{Y\sim q} \log(1-f(Y))\,, \quad (6)$$

and then the likelihood ratio is estimated as $f/(1-f)$. The last layer of the f networks are sigmoids in order to constrain their range to be between 0 and 1. The function f is additionally trained to be parameterized in θ by training with pairs (Y, Θ) instead of just Y, where Θ is a random variable corresponding to values θ sampled from a prior. We will use a uniform prior when training the parameterized classifiers.

All neural networks are implemented using PyTorch (Paszke et al., 2019) and optimized with Adam (Kingma & Ba, 2014) with a learning rate of 0.001 and consist of three hidden layers with 50 nodes per layer. All intermediate layers use ReLU activation functions. Each network is trained for 10,000 epochs with early stopping using a patience of 10. The w_1 training uses a batch size of 100,000. The w_0 network is simultaneously optimized with θ and uses a batch size that is the full dataset, which corresponds to performing the fit in Equation (4) over all the data.

3. Higgs Boson Cross Section

We now demonstrate the unfolding method in a physics case — a Higgs boson cross section measurement. Here, we focus on the di-photon decay channel of the Higgs boson. The goal is then to measure the transverse momentum spectrum of the Higgs boson p_H^{T} using the transverse momentum of the di-photon system $p_{\gamma\gamma}^{\text{T}}$ at detector level. The photon resolution ϵ_γ is considered as a nuisance parameter. In this case, the $p_{\gamma\gamma}^{\text{T}}$ spectrum is minimally affected by ϵ_γ. Therefore, we also consider the invariant mass spectrum of the di-photon system $m_{\gamma\gamma}$ at detector level, which is highly sensitive to ϵ_γ. In addition, In order to have a large spectrum difference between different data sets for demonstration purpose, we consider only events that contain at least two reconstructed jets, where the leading-order (LO) calculation would significantly differ from next-to-leading-order calculation (NLO)

We prepare the following data sets:

- D_{obs}: used as the observed data.
- $D_{\text{sim}}^{1.0}$: used as the nominal simulation sample.
- $D_{\text{sim}}^{1.2}$: used as the simulation sample with a systematic variation.
- D_{sim}^{*}: simulation sample with various ϵ_γ values for training the w_1 reweighter.

D_{obs} is generated at NLO using the POWHEGBOX program (Oleari, 2010; Alioli et al., 2009), while the rest are generated at LO using MADGRAPH5_aMC@LO v2.6.5 (Alwall et al., 2014). For all samples, the parton-level events are processed by PYTHIA 8.235 (Sjöstrand et al., 2006; 2015) for the Higgs decay, the parton shower, hadronization, and the underlying event. The detector simulation is based on DELPHES 3.5.0 (de Favereau et al., 2014) with detector response modified from the default ATLAS detector card. For both D_{obs} and $D_{\text{sim}}^{1.2}$, the photon resolution ϵ is multiplied by a factor of 1.2. For D_{sim}^{*}, the multiplier of ϵ is uniformly scanned between 0.5 and 1.5 with a step size of 0.01. $D_{\text{sim}}^{1.0}$ uses the default ATLAS detector card.

Each of the spectra of particle-level $p_{\gamma\gamma}^{\text{T}}$, detector-level $p_{\gamma\gamma}^{\text{T}}$ and detector-level $m_{\gamma\gamma}$ is standardized to the spectrum with a mean of 0 and a standard deviation of 1 before being passed to the neural networks. A w_1 reweighter is trained to reweight $D_{\text{sim}}^{1.0}$ to D_{sim}^{*}. The w_0 reweighter and ϵ are optimized simultaneously based on the pre-trained w_1 reweighter. The prior constraint of ϵ_γ is 50%. The fitted ϵ_γ is 1.19 ± 0.007. As shown in Figure 1, the reweighted detector-level spectra match well with observed data. The w_0 is then used to reweight the particle-level spectrum. As shown in Figure 2, the reweighted particle-level spectrum agrees with the truth (corresponds to observed data). This means that the observed data p_H^{T} spectrum is successfully unfolded with nuisance parameter ϵ_γ properly profiled. For comparison, we also perform UPU with ϵ_γ fixed at 1. As shown in Figure 2, the unfolded p_H^{T} spectrum in this case has a larger non-closure with the truth due to the lack of profiling.

4. Conclusion and Outlook

In this paper, we proposed Unbinned Profiled Unfolding (UPU), a new ML-based unfolding method that can process unbinned data and profile. The method uses the binned maximum likelihood as the figure of merit to optimize the unfolding reweighting function $w_0(t)$, which takes unbinned particle-level spectra as inputs. $w_0(t)$ and the nuisance parameters θ are optimized simultaneously, which also requires to learn a conditional likelihood ratio $w_1(t, r|\theta)$ that reweights the detector-level spectra based on the profiled values of nuisance parameters and is taken as an input for the optimization of $w_0(t)$ and θ.

We applied UPU to the Higgs boson cross section measurement. We considered one dimension at particle level and

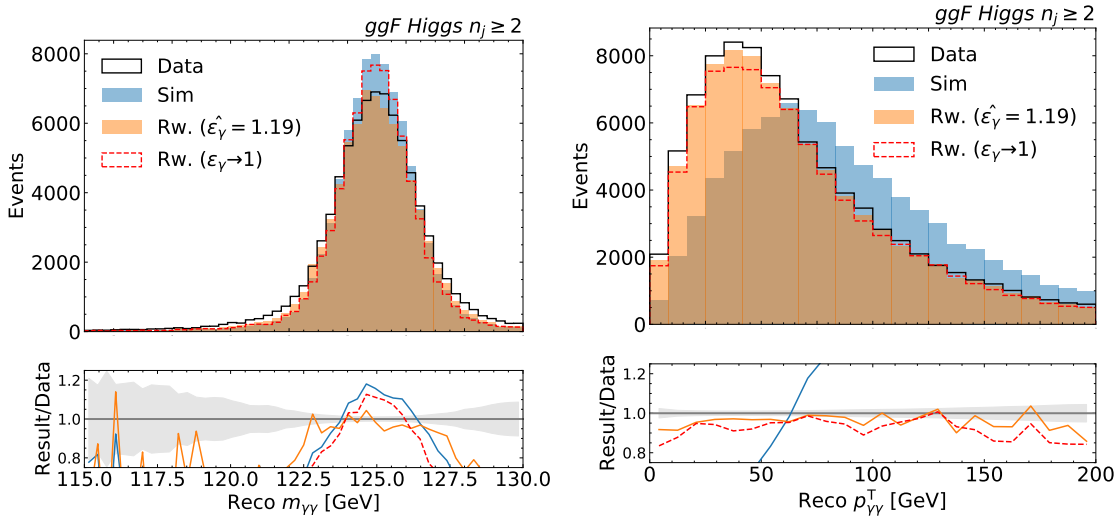

Figure 1. Higgs boson cross section: results of the w_0 optimization. The nuisance parameter ϵ_γ is optimized simultaneously with w_0 with the prior constraint set to 50% (orange) or fixed to 1 for comparison (red). The fitted ϵ_γ is 1.19 ± 0.007. (Left) The detector-level spectrum $m_{\gamma\gamma}$ of the simulation template D_{sim} reweighted by the trained $w_0 \times w_1$, compared to the $m_{\gamma\gamma}$ spectrum of the observed data D_{obs}. (Right) The detector-level spectrum $p_{\gamma\gamma}^{\mathrm{T}}$ of the simulation template D_{sim} reweighted by the trained $w_0 \times w_1$, compared to the $p_{\gamma\gamma}^{\mathrm{T}}$ spectrum of the observed data D_{obs}.

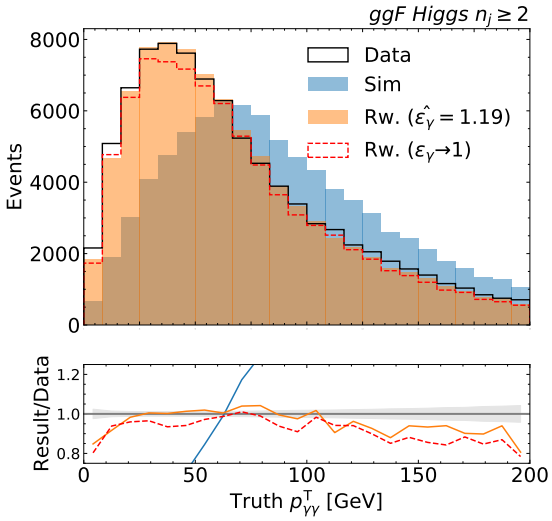

Figure 2. Higgs boson cross section: the particle-level spectrum $p_{\gamma\gamma}^{\mathrm{T}}$ of the simulation template D_{sim} reweighted by the trained w_0, compared to the $p_{\gamma\gamma}^{\mathrm{T}}$ spectrum of the observed data D_{obs}. The nuisance parameter ϵ_γ is optimized simultaneously with w_0 with the prior constraint set to 50% (orange) or fixed to 1 for comparison (red). The fitted ϵ_γ is 1.19 ± 0.007.

two dimensions at detector level. With one detector-level variable sensitive to the target particle-level observable and one sensitive to the effect of nuisance parameters, the data are successfully unfolded and profiled. The impact of profiling is also demonstrated by comparing with the result of nuisance parameter fixed to the nominal value. This can be readily extended to higher dimensions in either particle level or detector level, provided all particle-level and detector-level effects are distinguishable in the considered detector-level spectra. In the case of more than one nuisance parameters, one can either train multiple w_1 for each nuisance parameter separately or train a single w_1 which takes all nuisance parameters as inputs. As the effects of multiple nuisance parameters are usually assumed independent, one could take a product of individually trained reweighters.

As with any measurement, quantifying the uncertainty is critical to interpret UPU results. Just as in the binned case, one can calculate the uncertainty on the nuisance parameters which can be determined by fixing a given parameter to target values and then simultaneously re-optimizing w_0 and the rest of the nuisance parameters. A new feature of UPU is that the likelihood (ratio) itself is only an approximation, using neural networks as surrogate models. This is a challenge for all machine learning-based unfolding, and uncertainties can be probed by comparing the results with different simulations. Future extensions of UPU may be able to also use machine learning to quantify these model uncertainties as well as process unbinned data also at detector level.

Code and data

The code for this paper can be found at https://github.com/jaychan-hep/UnbinnedProfiledUnfolding, which uses Jupyter notebooks (Kluyver et al., 2016) and employs NumPy (Harris et al., 2020) for data manipulation and Matplotlib (Hunter, 2007) for visualization. The physics data sets are hosted on Zenodo at (Chan & Nachman, 2023).

Broader impact

The development of the Unbinned Profiled Unfolding (UPU) method presents significant potential for positive broader impacts in both the field of particle physics and society at large. By enabling unbinned unfolding and simultaneous profiling of nuisance parameters, this novel machine learning approach offers improved accuracy and precision in reconstructing particle energy spectra. The method's potential benefits include enhancing our understanding of fundamental particles and their interactions, thereby advancing scientific knowledge and contributing to the development of more precise experimental techniques in particle physics research.

Furthermore, UPU has the potential to extend beyond particle physics, with applications in various domains that involve unfolding problems and the estimation of hidden distributions from observed data. For instance, it could find applications in medical imaging, where accurate reconstruction of complex image structures from noisy and limited data is crucial. The method may also have broader implications in fields such as finance, environmental sciences, and social sciences, where accurate estimation of underlying distributions and profiling of influential factors are essential for decision-making processes.

References

Alioli, S., Nason, P., Oleari, C., and Re, E. NLO Higgs boson production via gluon fusion matched with shower in POWHEG. *JHEP*, 04:002, 2009. doi: 10.1088/1126-6708/2009/04/002.

Alwall, J., Frederix, R., Frixione, S., Hirschi, V., Maltoni, F., Mattelaer, O., Shao, H. S., Stelzer, T., Torrielli, P., and Zaro, M. The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations. *JHEP*, 07:079, 2014. doi: 10.1007/JHEP07(2014)079.

Amoroso, S. et al. Les Houches 2019: Physics at TeV Colliders: Standard Model Working Group Report. In *11th Les Houches Workshop on Physics at TeV Colliders: PhysTeV Les Houches*, 3 2020.

Andersen, J. R. et al. Les Houches 2015: Physics at TeV Colliders Standard Model Working Group Report. In *9th Les Houches Workshop on Physics at TeV Colliders*, 5 2016.

Andreassen, A., Komiske, P. T., Metodiev, E. M., Nachman, B., and Thaler, J. OmniFold: A Method to Simultaneously Unfold All Observables. *Phys. Rev. Lett.*, 124(18): 182001, 2020. doi: 10.1103/PhysRevLett.124.182001.

Andreassen, A., Komiske, P. T., Metodiev, E. M., Nachman, B., Suresh, A., and Thaler, J. Scaffolding Simulations with Deep Learning for High-dimensional Deconvolution. In *9th International Conference on Learning Representations*, 5 2021.

Arratia, M. et al. Publishing unbinned differential cross section results. *JINST*, 17(01):P01024, 2022. doi: 10.1088/1748-0221/17/01/P01024.

Backes, M., Butter, A., Dunford, M., and Malaescu, B. An unfolding method based on conditional Invertible Neural Networks (cINN) using iterative training. 12 2022.

Balasubramanian, R., Brenner, L., Burgard, C., Cowan, G., Croft, V., Verkerke, W., and Verschuuren, P. Statistical method and comparison of different unfolding techniques using RooFit. 2019.

Bellagente, M., Butter, A., Kasieczka, G., Plehn, T., and Winterhalder, R. How to GAN away Detector Effects. 2019.

Bellagente, M., Butter, A., Kasieczka, G., Plehn, T., Rousselot, A., Winterhalder, R., Ardizzone, L., and Köthe, U. Invertible Networks or Partons to Detector and Back Again. *SciPost Phys.*, 9:074, 2020. doi: 10.21468/SciPostPhys.9.5.074.

Berger, N. et al. Simplified Template Cross Sections - Stage 1.1. 6 2019.

Blobel, V. Unfolding Methods in Particle Physics. *PHYSTAT2011 Proceedings*, pp. 240, 2011. doi: 10.5170/CERN-2011-006.

Blobel, V. Unfolding. *Data Analysis in High Energy Physics*, pp. 187, 2013. doi: 10.1002/9783527653416.ch6. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/9783527653416.ch6.

Chan, J. and Nachman, B. Higgs to diphoton channel at least 2 jet datasets, January 2023. URL https://doi.org/10.5281/zenodo.7553271.

Choudalakis, G. Fully bayesian unfolding, 2012. URL https://arxiv.org/abs/1201.4612.

Cowan, G. A survey of unfolding methods for particle physics. *Conf. Proc.*, C0203181:248, 2002.

D'Agostini, G. A Multidimensional unfolding method based on Bayes' theorem. *Nucl. Instrum. Meth.*, A362:487–498, 1995. doi: 10.1016/0168-9002(95)00274-X.

Datta, K., Kar, D., and Roy, D. Unfolding with Generative Adversarial Networks. 2018.

de Favereau, J., Delaere, C., Demin, P., Giammanco, A., Lemaître, V., Mertens, A., and Selvaggi, M. DELPHES 3, A modular framework for fast simulation of a generic collider experiment. *JHEP*, 02:057, 2014. doi: 10.1007/JHEP02(2014)057.

de Florian, D. et al. Handbook of LHC Higgs Cross Sections: 4. Deciphering the Nature of the Higgs Sector. 2/2017, 10 2016. doi: 10.23731/CYRM-2017-002.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pp. 2672–2680, Cambridge, MA, USA, 2014. MIT Press. URL http://dl.acm.org/citation.cfm?id=2969033.2969125.

H1 Collaboration. Measurement of Lepton-Jet Correlation in Deep-Inelastic Scattering with the H1 Detector Using Machine Learning for Unfolding. *Phys. Rev. Lett.*, 128(13):132002, 2022a. doi: 10.1103/PhysRevLett.128.132002.

H1 Collaboration. Machine learning-assisted measurement of multi-differential lepton-jet correlations in deep-inelastic scattering with the H1 detector. *H1prelim-22-031*, 2022b. URL https://www-h1.desy.de/h1/www/publications/htmlsplit/H1prelim-22-031.long.html.

H1 Collaboration. Multi-differential Jet Substructure Measurement in High Q^2 DIS Events with HERA-II Data. *H1prelim-22-034*, 2022c. URL https://www-h1.desy.de/h1/www/publications/htmlsplit/H1prelim-22-034.long.html.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

Hastie, T., Tibshirani, R., and Friedman, J. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

Heinrich, L., Feickert, M., and Stark, G. pyhf: v0.7.0. URL https://doi.org/10.5281/zenodo.1169739. https://github.com/scikit-hep/pyhf/releases/tag/v0.7.0.

Heinrich, L., Feickert, M., Stark, G., and Cranmer, K. pyhf: pure-python implementation of histfactory statistical models. *Journal of Open Source Software*, 6 (58):2823, 2021. doi: 10.21105/joss.02823. URL https://doi.org/10.21105/joss.02823.

Hocker, A. and Kartvelishvili, V. SVD approach to data unfolding. *Nucl. Instrum. Meth.*, A372:469–481, 1996. doi: 10.1016/0168-9002(95)01478-0.

Howard, J. N., Mandt, S., Whiteson, D., and Yang, Y. Foundations of a Fast, Data-Driven, Machine-Learned Simulator. 1 2021.

Hunter, J. D. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization, 2014. URL https://arxiv.org/abs/1412.6980.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. 2014.

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., and Willing, C. Jupyter notebooks – a publishing format for reproducible computational workflows. In Loizides, F. and Schmidt, B. (eds.), *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pp. 87 – 90. IOS Press, 2016.

LHCb Collaboration. Multidifferential study of identified charged hadron distributions in Z-tagged jets in proton-proton collisions at $\sqrt{s}$ =13 TeV. *arXiv:2208.11691*, 8 2022.

Nachman, B. and Thaler, J. Neural Conditional Reweighting. *Phys. Rev. D*, 105:076015, 2022. doi: 10.1103/PhysRevD.105.076015.

Oleari, C. The POWHEG-BOX. *Nucl. Phys. B Proc. Suppl.*, 205-206:36–41, 2010. doi: 10.1016/j.nuclphysbps.2010.08.016.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N.,

Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Rezende, D. J. and Mohamed, S. Variational inference with normalizing flows. *International Conference on Machine Learning*, 37:1530, 2015.

Schmitt, S. TUnfold: an algorithm for correcting migration effects in high energy physics. *JINST*, 7:T10003, 2012. doi: 10.1088/1748-0221/7/10/T10003.

Sjöstrand, T., Mrenna, S., and Skands, P. Z. PYTHIA 6.4 Physics and Manual. *JHEP*, 05:026, 2006. doi: 10.1088/1126-6708/2006/05/026.

Sjöstrand, T., Ask, S., Christiansen, J. R., Corke, R., Desai, N., Ilten, P., Mrenna, S., Prestel, S., Rasmussen, C. O., and Skands, P. Z. An Introduction to PYTHIA 8.2. *Comput. Phys. Commun.*, 191:159–177, 2015. doi: 10.1016/j.cpc.2015.01.024.

Sugiyama, M., Suzuki, T., and Kanamori, T. *Density Ratio Estimation in Machine Learning*. Cambridge University Press, 2012. doi: 10.1017/CBO9781139035613.

Vandegar, M., Kagan, M., Wehenkel, A., and Louppe, G. Neural Empirical Bayes: Source Distribution Estimation and its Applications to Simulation-Based Inference. In Banerjee, A. and Fukumizu, K. (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 2107–2115. PMLR, 11 2021. URL https://proceedings.mlr.press/v130/vandegar21a.html.

A. Results of the w_1 training for the Higgs Boson Cross Section measurement case

As described in Section 3, the trained w_1 is tested with the nominal detector level $p_{\gamma\gamma}^{\mathrm{T}}$ and $m_{\gamma\gamma}$ spectra ($D_{\mathrm{sim}}^{1.0}$) reweighted to $\epsilon_\gamma = 1.2$ and compared to the detector level $p_{\gamma\gamma}^{\mathrm{T}}$ and $m_{\gamma\gamma}$ spectra with $\epsilon_\gamma = 1.2$. As shown in Figure 3, the trained w_1 reweighter has learned to reweight the nominal detector level $m_{\gamma\gamma}$ spectrum to match the detector level $m_{\gamma\gamma}$ spectrum with ϵ_γ at 1.2, and the detector level $p_{\gamma\gamma}^{\mathrm{T}}$ variable is independent of the w_1 reweighter.

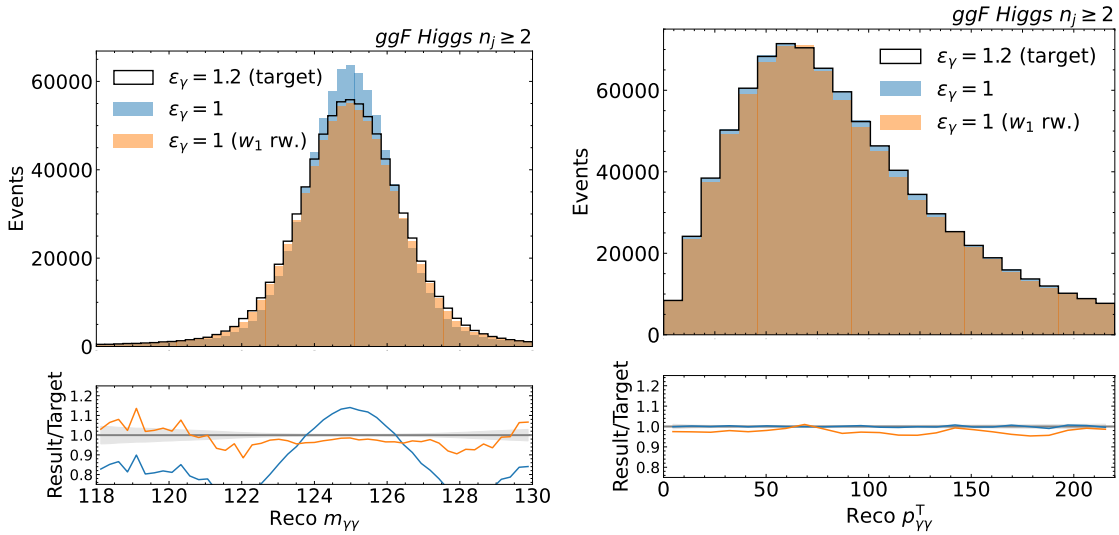

Figure 3. Higgs boson cross section: the nominal detector-level spectra $m_{\gamma\gamma}$ (left) and $p_{\gamma\gamma}^{\mathrm{T}}$ (right) with $\epsilon_\gamma = 1$ reweighted by the trained w_1 conditioned at $\epsilon_\gamma = 1.2$ and compared to the spectra with $\epsilon_\gamma = 1.2$.

B. Gaussian Examples

B.1. One-dimension in both particle and detector level

In this appendix, we demonstrate the proposed method with simple numerical examples. Here, each data set represents one-dimensional Gaussian random variables in both the particle and detector level. The particle-level random variable T is described by mean μ and standard deviation σ, while the detector-level variable is given by

$$R = T + Z, \tag{7}$$

where Z is a Gaussian random variable with mean β and standard deviation ϵ.

In a first example, ϵ is considered to be the only nuisance parameter, and β is fixed to 0. Three data sets are prepared for the full training procedure. The first data set $D_{\mathrm{sim}}^{1.0}$ is used as the nominal simulation sample, which contains 200,000 events with $\mu = 0$, $\sigma = 1$ and $\epsilon = 1$. The second data set D_{obs} is used as the observed data, which contains 100,000 events with $\mu = 0.2$, $\sigma = 1$ and $\epsilon = 1.2$. To train the w_1 reweighter, the third data set D_{sim}^{*}, which contains 200,000 events with $\mu = 0$, $\sigma = 1$ and ϵ uniformly distributed from 0.2 to 1.8, is prepared. In addition, another data set $D_{\mathrm{sim}}^{1.2}$ of 100,000 events with $\mu = 0$, $\sigma = 1$ and $\epsilon = 1.2$ is produced for validating the w_1 reweighter. All data sets used in the training procedure are split to 50% for training and 50% for validating.

A w_1 reweighter is trained to reweight $D_{\mathrm{sim}}^{1.0}$ to D_{sim}^{*}. The trained w_1 is then tested with the nominal R distribution ($D_{\mathrm{sim}}^{1.0}$) reweighted to $\epsilon = 1.2$ ($w_1\,(R|T, \epsilon = 1.2)$) and compared to the R spectrum with $\epsilon = 1.2$ ($D_{\mathrm{sim}}^{1.2}$). As shown in Figure 4, the trained w_1 reweighter has learned to reweight the nominal R spectrum to match the R spectrum with ϵ at 1.2.

With this trained w_1 reweighter, a w_0 reweighter is trained using $D_{\mathrm{sim}}^{1.0}$ as the simulation template with D_{obs} as the observed data used in Equation (4). In the first scenario, the nuisance parameter ϵ for the w_1 reweighter is fixed to 1.2, and the penalty term in Equation (4) $\log(\theta)$ is set to 0 (no constraint). As shown in Figure 5, the w_0 reweighter is able to learn to reweight the particle-level spectrum T by matching the detector-level spectrum R to the observed spectrum. In the second scenario,

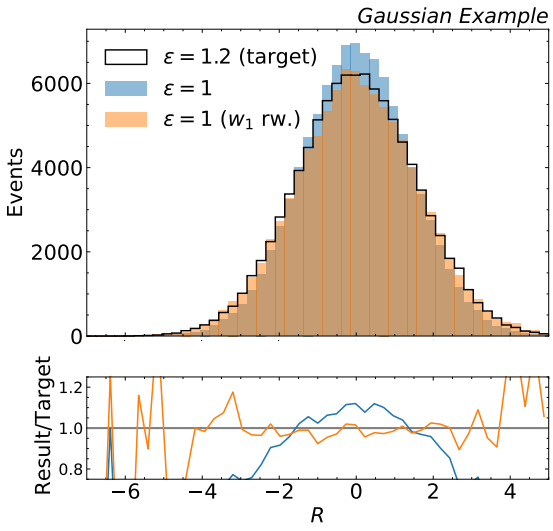

Figure 4. Gaussian 1D example: the nominal R distribution ($\epsilon = 1$) in the reweighted by the trained w_1 conditioned at $\epsilon = 1.2$ and compared to R distribution with $\epsilon = 1.2$.

the nuisance parameter ϵ is trained together with the w_0 reweighter. The prior in the penalty term in Equation (4) is set to be a Gaussian probability density with a 80% uncertainty. As shown in Figure 6, the trained w_0 and optimized ϵ are tested. The fitted ϵ is 1.03 ± 0.016 [1] (true value is 1.2). The reweighted distribution matches well with observed data in the detector-level spectrum but the particle-level spectrum has a large non-closure. This is because of the degeneracy between the w_0 and w_1 reweighters in the effect on the detector-level spectrum. In other words, detector effects can mimic changes in the particle-level cross section, so the data cannot distinguish between these two scenarios. This is a common issue which also exists in the standard binned maximum likelihood unfolding. For comparison, we also perform the standard binned maximum likelihood unfolding. As shown in Appendix C, the unfolded T spectrum in this case also fails to represent the true T spectrum. An 80% uncertainty is highly exaggerated from typical scenarios, but it clearly illustrates the challenge of profiling and unfolding at the same time (see Section 4 for a discussion about regularization).

B.2. One-dimension in particle level and two-dimension in detector level

To break the degeneracy between the w_0 and w_1 reweighters, we now consider a two-dimension distribution in the detector level, which is given by

$$R = T + Z, \tag{8}$$
$$R^* = T + Z^*, \tag{9}$$

where Z (Z^*) is a Gaussian random variable with mean β (β^*) and standard deviation ϵ (ϵ^*). ϵ is considered to be the only nuisance parameter, and β, β^* are fixed to 0, and ϵ^* is fixed to 1. In this case, the nuisance parameter ϵ only has effect on the R spectrum and the R^* spectrum depends purely on the particle-level spectrum T.

Similar to the previous example, $D^{1.0}_{\text{sim}}$ is used as the nominal simulation sample, which contains 200,000 events with $\mu = 0$, $\sigma = 1$ and $\epsilon = 1$. D_{obs} is used as the observed data, which contains 100,000 events with $\mu = 0.8$, $\sigma = 1$ and $\epsilon = 1.2$. To train the w_1 reweighter, D^*_{sim}, which contains 200,000 events with $\mu = 0$, $\sigma = 1$ and ϵ uniformly distributed from 0.2 to 1.8, is prepared. In addition, another data set $D^{1.2}_{\text{sim}}$ of 100,000 events with $\mu = 0$, $\sigma = 1$ and $\epsilon = 1.2$ is produced for validating the w_1 reweighter. All data sets used in the training procedure are split to 50% for training and 50% for validating.

A w_1 reweighter is trained to reweight $D^{1.0}_{\text{sim}}$ to D^*_{sim}. The trained w_1 is tested with the nominal R and R^* spectra ($D^{1.0}_{\text{sim}}$) reweighted to $\epsilon = 1.2$ and compared to the R and R^* spectra with $\epsilon = 1.2$. As shown in Figure 7, the trained w_1 reweighter has learned to reweight the nominal R spectrum to match the R spectrum with ϵ at 1.2, and R^* is independent of the w_1

[1] The fitted value is averaged over five different w_0 reweighters which are trained in the same way, but with different random initializations. The standard deviation of the fitted values is taken as the error.

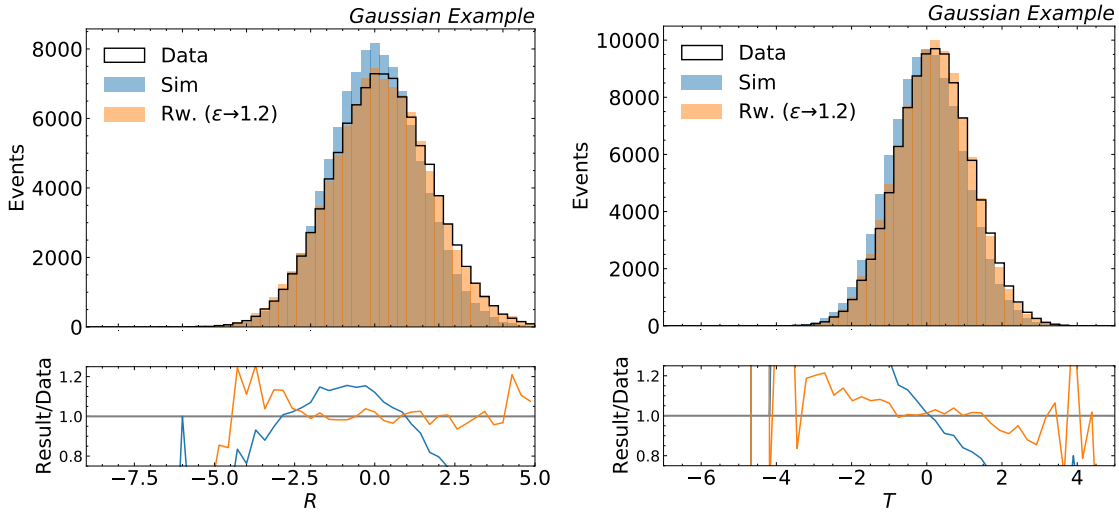

Figure 5. Gaussian 1D example: results of the w_0 optimization. The nuisance parameter ϵ is fixed to 1.2, and the the penalty term in Equation (4) is set to 0. (Top) The detector-level spectrum R of the simulation template D_{sim} reweighted by the trained $w_0 \times w_1$, compared to the R spectrum of the observed data D_{obs}. (Bottom) The particle-level spectrum T of the simulation template D_{sim} reweighted by the trained w_0, compared to the T spectrum of the observed data D_{obs}.

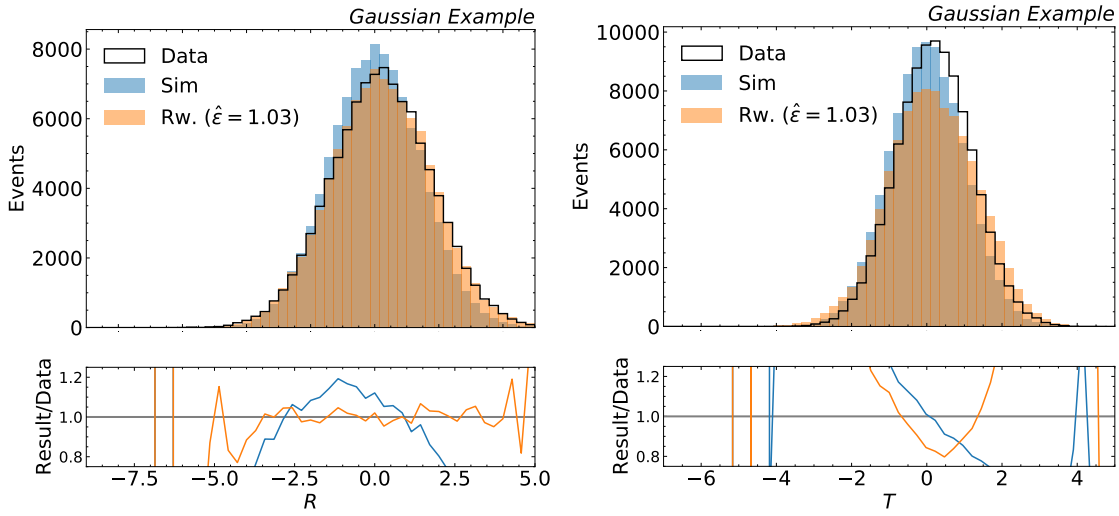

Figure 6. Gaussian 1D example: results of the w_0 optimization. The nuisance parameter ϵ is optimized simultaneously with w_0 and the best-fit value is $\hat{\epsilon} = 1.03 \pm 0.016$. (Left) The detector-level spectrum R of the simulation template D_{sim} reweighted by the trained $w_0 \times w_1$, compared to the R spectrum of the observed data D_{obs}. (Right) The particle-level spectrum T of the simulation template D_{sim} reweighted by the trained w_0, compared to the T spectrum of the observed data D_{obs}.

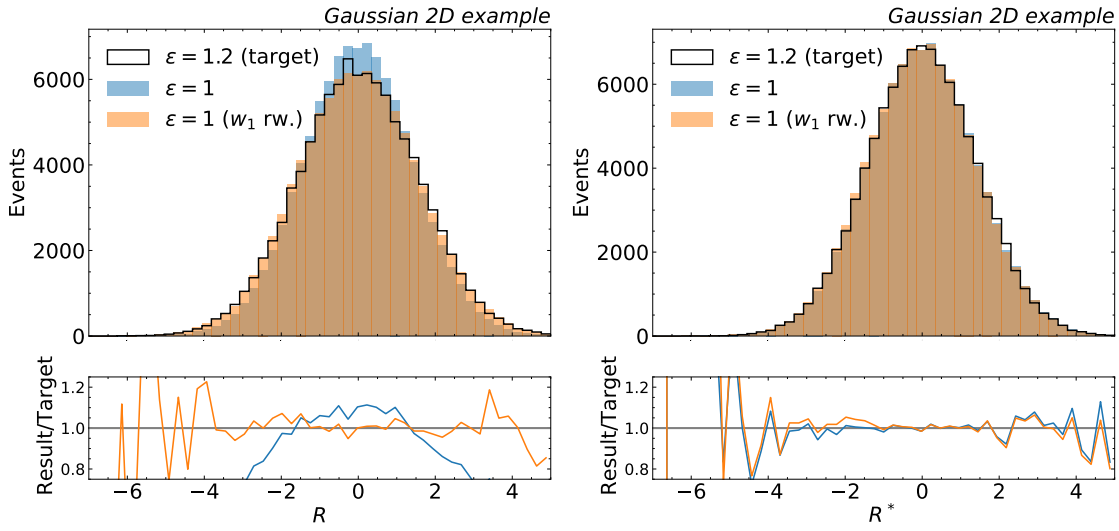

Figure 7. Gaussian 2D example: the nominal detector-level spectra R (left) and R^* (right) with $\epsilon = 1$ reweighted by the trained w_1 conditioned at $\epsilon = 1.2$ and compared to the spectra with $\epsilon = 1.2$.

reweighter.

Based on the trained w_1 reweighter, a w_0 reweighter and the nuisance parameter ϵ are optimized simultaneously using D_{sim} as the simulation template with D_{obs} as the observed data used in Equation (4). As before, the prior in the penalty term in Equation (4) is configured with an uncertainty of 80%. The fitted ϵ is 1.20 ± 0.004 (correct value is 1.2). As shown in Figure 8, the reweighted spectra match well with observed data in both detector and particle level. For more realistic uncertainties (so long as the simulation is close to the right answer), the fidelity is even better.

C. Binned maximum likelihood unfolding with Gaussian examples

In this appendix, we present results of the standard binned maximum likelihood unfolding (BMLU) with Gaussian examples. The scenarios are:

- One-dimension in both particle and detector level: this is the same example as described in Appendix B.1. The prior constraint for ϵ is set to 80%. The result is shown in Figure 9 with ϵ fitted to 1.08 ± 0.02, which also indicates a degeneracy problem between particle and detector levels.
- One-dimension in particle level and two-dimension in detector level: this is the same example as described in Appendix B.2. The prior constraint for ϵ is set to 80%. The result is shown in Figure 10 with ϵ fitted to 1.19 ± 0.003. The degeneracy problem is resolved after considering an additional spectrum in the detector level.

  All the maximum likelihood fittings are performed using pyhf (Heinrich et al.; 2021).

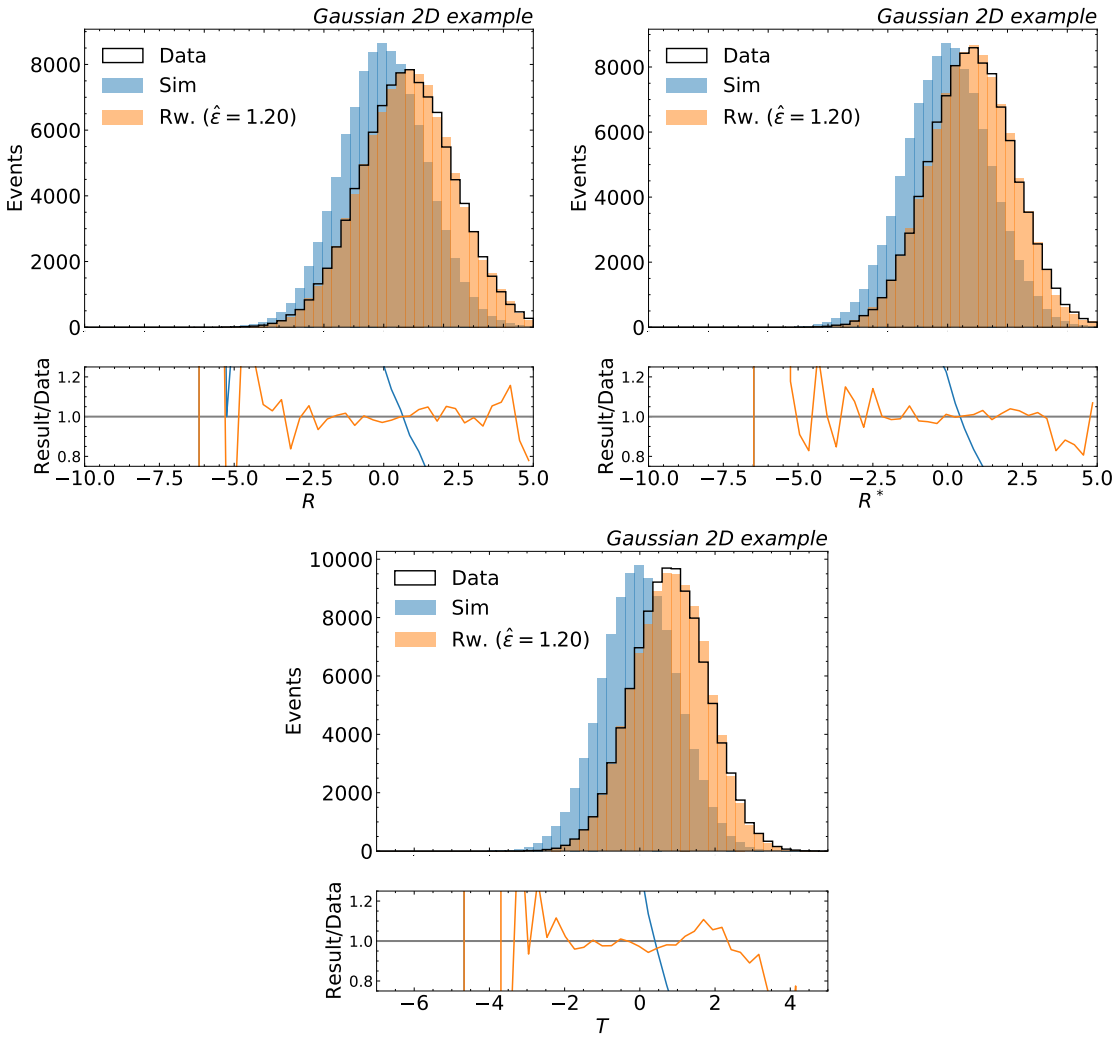

Figure 8. Gaussian 2D example: results of the w_0 optimization. The nuisance parameter ϵ is optimized simultaneously with w_0 with the prior constraint set to 80%. The fitted ϵ is 1.20 ± 0.004. (Top-left) The detector-level spectrum R of the simulation template D_{sim} reweighted by the trained $w_0 \times w_1$, compared to the R spectrum of the observed data D_{obs}. (Top-right) The detector-level spectrum R' of the simulation template D_{sim} reweighted by the trained $w_0 \times w_1$, compared to the R^* spectrum of the observed data D_{obs}. (Bottom) The particle-level spectrum T of the simulation template D_{sim} reweighted by the trained w_0, compared to the T spectrum of the observed data D_{obs}.

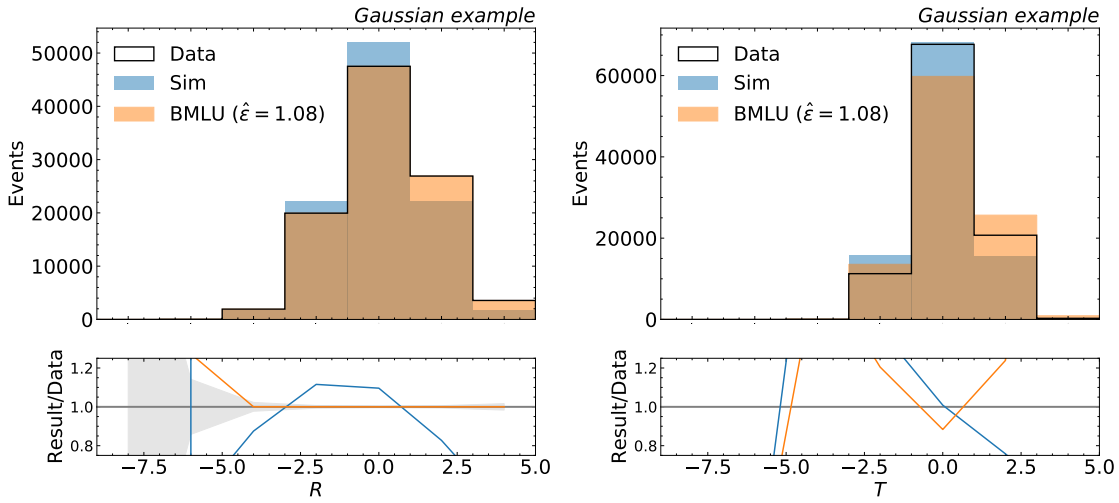

Figure 9. Gaussian 1D example: results of the binned maximum likelihood unfolding. The prior constraint for ϵ is set to 80% and the fitted ϵ is 1.08 ± 0.02. (Left) The fitted detector-level spectrum R of the simulation template D_{sim}, compared to the R spectrum of the observed data D_{obs}. (Right) The unfolded particle-level spectrum T of the simulation template D_{sim}, compared to the T spectrum of the observed data D_{obs}.

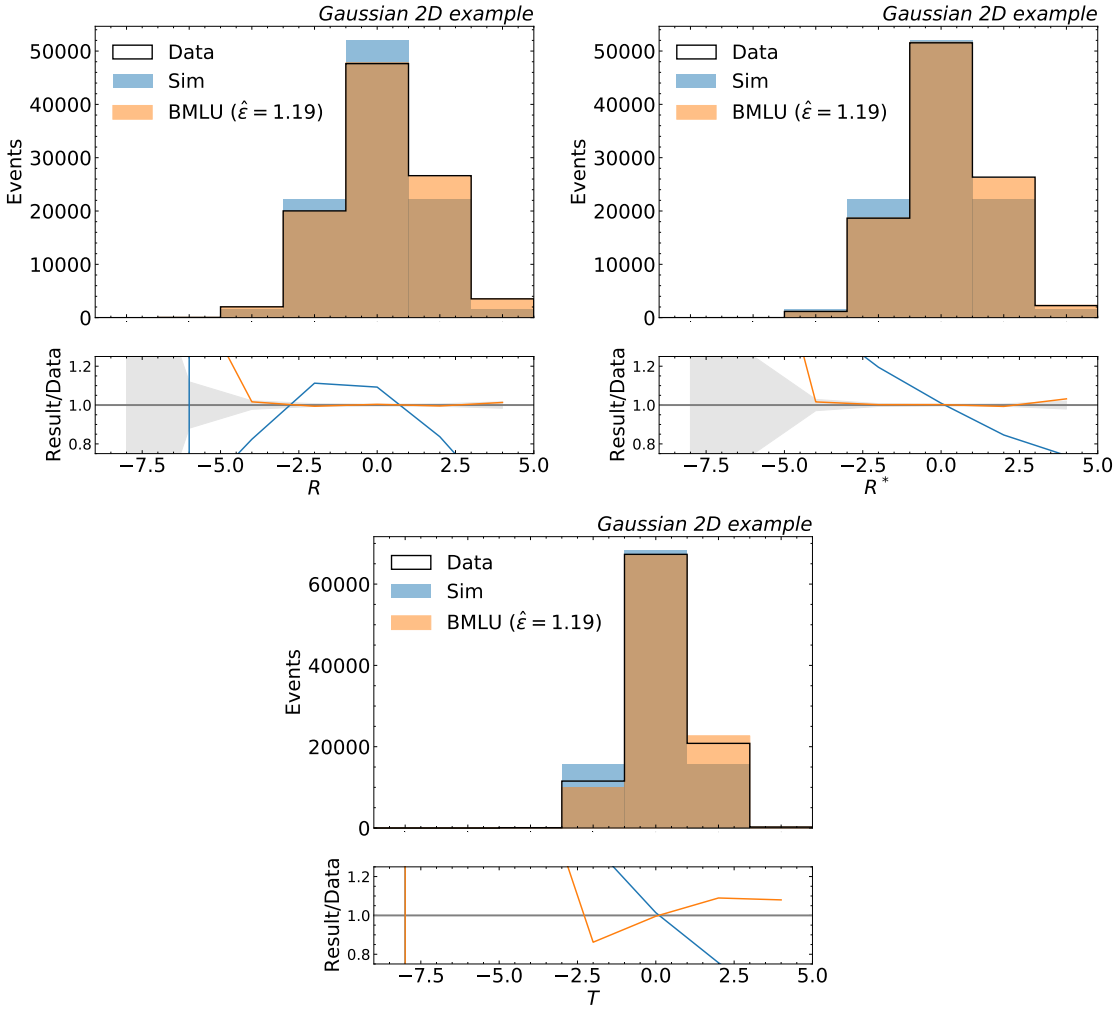

Figure 10. Gaussian 2D example: results of the binned maximum likelihood unfolding. The prior constraint for ϵ is set to 80% and the fitted ϵ is 1.19 ± 0.003. (Top-left) The fitted detector-level spectrum R of the simulation template D_{sim}, compared to the R spectrum of the observed data D_{obs}. (Top-right) The fitted detector-level spectrum R^* of the simulation template D_{sim}, compared to the R' spectrum of the observed data D_{obs}. (Bottom) The unfolded particle-level spectrum T of the simulation template D_{sim}, compared to the T spectrum of the observed data D_{obs}.