Published for SISSA by Springer

Received: September 12, 2023
Revised: February 6, 2024
Accepted: February 26, 2024
Published: March 20, 2024

Boosting likelihood learning with event reweighting

Siyu Chen,[a] Alfredo Glioti,[b] Giuliano Panico[c,d] and Andrea Wulzer[e,f]

[a]Institut de Théorie des Phénomenes Physiques, EPFL,
Lausanne, Switzerland
[b]Université Paris-Saclay, CNRS, CEA, Institut de Physique Théorique,
91191, Gif-sur-Yvette, France
[c]Dipartimento di Fisica e Astronomia, Università di Firenze,
Via G. Sansone 1, 50019 Sesto Fiorentino, Italy
[d]INFN, Sezione di Firenze,
Via G. Sansone 1, 50019 Sesto Fiorentino, Italy
[e]Institut de Física d'Altes Energies (IFAE),
The Barcelona Institute of Science and Technology (BIST),
Campus UAB, 08193 Bellaterra, Barcelona, Spain
[f]ICREA, Institució Catalana de Recerca i Estudis Avançats,
Passeig de Lluís Companys 23, 08010 Barcelona, Spain

E-mail: alfredo.glioti@ipht.fr, giuliano.panico@unifi.it, wulzer@ifae.es

Abstract: Extracting maximal information from experimental data requires access to the likelihood function, which however is never directly available for complex experiments like those performed at high energy colliders. Theoretical predictions are obtained in this context by Monte Carlo events, which do furnish an accurate but abstract and implicit representation of the likelihood. Strategies based on statistical learning are currently being developed to infer the likelihood function explicitly by training a continuous-output classifier on Monte Carlo events. In this paper, we investigate the usage of Monte Carlo events that incorporate the dependence on the parameters of interest by reweighting. This enables more accurate likelihood learning with less training data and a more robust learning scheme that is more suited for automation and extensive deployment. We illustrate these advantages in the context of LHC precision probes of new Effective Field Theory interactions.

Keywords: Electroweak Precision Physics, SMEFT

ArXiv ePrint: 2308.05704

https://doi.org/10.1007/JHEP03(2024)117

Contents

1 Introduction

The problem of extracting maximal information from precise experimental measurements compared with precise theoretical predictions is of prime relevance in several domains of science. In the context of high-energy collider physics, the problem should be addressed for the exploitation of the data collected by the Large Hadron Collider (LHC) experiments and of those of its forthcoming High Luminosity (HL-LHC) upgrade. Precision physics is also a major component of proposed future collider projects such as an e^+e^- Higgs factory or a high-energy muon collider. The corresponding sensitivity projection studies can thus benefit from advances in precision physics analysis methodologies.

High energy physics data sets, $\mathcal{D} = \{x_i\}_{i=1}^{\mathcal{N}}$, consist of repeated measurements of a multi-component statistical variable x of observables. The number of data points $\mathcal{N}$ is also a statistical variable, which follows a Poisson distribution with expected N. Both the probability distribution of x and the expected number of events are controlled by the microscopic laws of fundamental interactions, which in turn can depend on a number of parameters of interest, c. Specifically, the differential cross section $d\sigma(x;c)$ depends on the parameters of interest and so it does the total cross section $\sigma(c)$, which is the integral of $d\sigma$ over the support of the variable x. The probability density function of x is the ratio $d\sigma(x;c)/\sigma(c)$. The total number of expected events is $N(c) = L\,\sigma(c)$, where the proportionality factor L — the integrated luminosity — is determined by the properties of the collider including its run time. The task of the analysis is to extract information on the parameters of interest by comparing the observed data with their parameter-dependent expected distribution.

The parameters of interest c could be free parameters of the currently established theoretical description of fundamental interactions, the Standard Model (SM) theory. Or, they could parametrize deformations of the SM due to additional or different interactions. In the latter case, which is by far the most common one for LHC and HL-LHC applications, the first goal of the analysis is to establish whether the data favor the SM point $c = 0$ in the parameter space, or if instead $c \neq 0$ is preferred hinting to non-SM fundamental physics laws. In the former case, one would like to set an exclusion limit, namely to quantify the maximal value that the c parameters can conceivably assume given that the data have not revealed their presence. In the latter case, one would first aim at quantifying the degree of confidence for the discovery of non-SM physical laws, i.e. the confidence level for SM exclusion. The measurement of the value of the c parameters will become relevant at a later stage.

Regardless of the specific goal of the analysis, any classical or Bayesian statistical inference methodology aimed at optimality[1] is based on the knowledge of the likelihood function associated with the experimental data. This is given in our case by the extended likelihood

$$\mathfrak{L}(c;\mathcal{D}) = \mathrm{Pois}[\mathcal{N}|N(c)]\prod_{x\in\mathcal{D}} p(x|c) = \frac{L^{\mathcal{N}}}{\mathcal{N}!}e^{-N(c)}\prod_{x\in\mathcal{D}} d\sigma(x;c)\,. \tag{1.1}$$

More precisely, what is truly needed is the dependence of the likelihood on the c parameters, up to a c-independent normalization factor. It is natural in our context to employ the likelihood at the SM point $c = 0$ for normalization. Optimal statistical inference can thus be attained by the knowledge of the likelihood log-ratio

$$\lambda(c;\mathcal{D}) \equiv \log\frac{\mathfrak{L}(c;\mathcal{D})}{\mathfrak{L}(0;\mathcal{D})} = N(0) - N(c) + \sum_{x\in\mathcal{D}} \log r(x;c)\,, \tag{1.2}$$

having defined the ratio of differential cross sections

$$r(x;c) = \frac{d\sigma(x;c)}{d\sigma(x;0)}\,. \tag{1.3}$$

As we will show, the knowledge of the likelihood log-ratio, allows one to optimize the sensitivity to new-physics in a point-by-point way in the c space. In practical terms, we propose to achieve this through a binned analysis in which the variable used for binning, as well as the bin boundaries, are chosen on the basis of the likelihood ratio for a given value of c. This is rather different from regular LHC binned exclusion analyses, in which the bins are pre-specified, so that no near-optimality is guaranteed.

In high-energy physics, the theoretical comprehension of fundamental interactions is translated into predictions for the outcome of experiments by a sophisticated chain of analytical and numerical tools that eventually produces Monte Carlo events. In the simplest setup, the events are *unweighted.* In this case, the events are statistical samples that follow the x variable distribution $d\sigma(x;c)/\sigma(c)$. An estimate of the total cross section $\sigma(c)$ — and in turn of the expected number of events $N(c)$ — is also available at the end of the event generation process

[1] The precise notion of optimality within a frequentist approach is defined in section 4.

as the result of the Monte Carlo integration. Equal weights are conventionally assigned to unweighted events, given by $\sigma(c)$ divided by the number of generated events. Notice that since the differential cross section depends on c, independent sets of Monte Carlo events need to be generated at different points in the c parameter space. This limitation can be avoided by employing instead *weighted* Monte Carlo events, to be described later.

Importantly enough, the Monte Carlo event generators in high energy physics do not sample from the distribution of x directly. The differential cross section $d\sigma(x;c)$ can never be computed theoretically and is unknown. What is instead computed and available is the differential cross section $d\sigma(\xi;c)$ in a space of latent variables ξ. The ξ variables are in general completely unrelated with the measured variables x. The Monte Carlo code generates samples in the ξ space and obtains events in the x space by propagating ξ samples through a series of steps that eventually entail dimensionality reduction. In fact, several of these steps — such as the QCD and QED radiation showering and the simulation of the detector response — are performed by randomized algorithms. The random variables they draw are effectively additional components of the large latent variable vector that is ultimately projected onto the space of observable variables x. It is normally impossible to model this complex process analytically such as to obtain a closed form for the distribution in the x space starting from the known distribution $d\sigma(\xi;c)$ in the latent space. One should perform multiple convolutions, and integrate over the unobserved components of the latent vector. These integrals can not be solved analytically and numerical approaches are not viable because the integration should be performed point-by-point in the x space. In this paper, we employ a simple Monte Carlo generator for the validation of our method in a fully controlled "ideal" setup, described in appendix A. This generator exemplifies in practice the role of latent variables. By design, our ideal setup enables a simple integration — by a finite sum, since the variables are discrete — over the latent space. In the realistic setups to be employed for real data analyses, it is on the contrary never possible to perform the integration. The x distribution can not be determined and consequently, we do not have access to the likelihood ratio (1.2). It is normally possible to determine $N(c) = L\,\sigma(c)$ by the Monte Carlo integration. What is missing is the distribution ratio $r(x;c)$ defined in eq. (1.3).

Several groups [1–13] recently investigated the possibility of extracting $r(x;c)$ from Monte Carlo simulations by employing statistical learning techniques. The goal is to address the limitations of traditional methods such as the matrix element method [14–20] and similar techniques [21–30]. In particular, the novel methodologies do not rely on approximate phenomenological modeling of the distribution. Hence, they promise to be universally applicable, simpler and faster to set up and to run, as well as systematically improvable by employing more accurate Monte Carlo generators and larger training data sets. They could be automated to a large extent, enabling their extensive deployment. Attempts are MadMiner [4] and ML4EFT [10].

A striking use case for the deployment of these methodologies at the LHC and beyond is the problem of testing new physics effects described by interaction operators of dimension larger than four in an Effective Field Theory (EFT) framework. Most of the work performed so far, and the one presented in the present paper, is specifically tailored to EFT applications. A simplification emerges in this context in that EFTs can be analyzed with 'theory aware' learning techniques. In particular one can use quadratic classifiers [5], that encode the analytic dependence of the differential cross sections on the EFT parameters.

Many of the results obtained in the EFT context, including ours, are however portable to other problems. In general, any physics effects that can be modeled through a simple analytic dependence on a set of parameters is amenable to 'theory aware' techniques. For instance, small systematic effects parametrized in terms of nuisance parameters at the linear order, among which PDF, scale and parton shower uncertainties, could be learned with analogous strategies.

In this paper, we demonstrate the advantages of learning $r(x;c)$ by training the statistical model with weighted Monte Carlo events, as obtained from generators that incorporate the dependence on the c parameters by the technique of *event reweighting*. Weighted Monte Carlo events, unlike unweighted ones, are not samples of the x variable, though they cover the same support with a similar distribution. They come with their own weights, to be employed for weighted sums in the calculation of expectation values. The sum of the weights in the data set is equal to the total cross section, and the sum of the weights of those events that fall in a certain bin (i.e., a region of the x variable space) equals the cross section in that bin. In general, the weighted sum of any observable $\mathcal{O}(x)$ evaluated over the events provides an estimate of the expectation value of the observable over x, multiplied by the total cross section. It is often merely a matter of convenience whether to employ weighted or unweighted events.[2] Weighted events obtained by reweighting are more useful in our context.

Event reweighting exploits the knowledge of the differential cross section $d\sigma(\xi;c)$ in the latent space. The generator code has access to this function, as well as to the value of the ξ variables for each event that is sampled. After the sampling in the ξ space, all the steps performed by the generator code — including QCD and QED radiation showering and detector simulation, and the projection from the latent to the x space — are independent of the value assumed by the c parameters. One can thus proceed as follows. First, generate a set of Monte Carlo events at the point $c = 0$ in the parameter space. We denote as $w_{\rm e}(0)$ the weight of each event "e" in this set. The $c = 0$ data set could be an unweighted one, in which case the $w_{\rm e}(0)$ weights are all equal, or be a weighted set with non-trivial $w_{\rm e}(0)$ weights. Next, run through the list of generated events and assign them a weight $w_{\rm e}(c) = w_{\rm e}(0)\, d\sigma(\xi;c)/d\sigma(\xi;0)$. These weights account for the dependence on c of the ξ sampling, which in turn captures the entire dependence on c of the observables because all the subsequent steps of the generation are independent of c. The single event data set $\mathrm{S} = \{\mathrm{e}_i\}_{i=1}^{n(\mathrm{S})}$ — where each event is a pair $\mathrm{e} = (x_{\rm e},\, w_{\rm e}(c))$ — generated by reweighting a single run of the Monte Carlo code with c equal to zero, as previously described, is thus a valid weighted set that enables the prediction of expectation values for arbitrary c.

It must be noticed that the use of weighted events can have some drawbacks on the accuracy of the Monte Carlo simulations, which need to be carefully monitored. On the one hand, reweighting relies on the assumption that all the relevant kinematic regions are populated by the original data-set, so that reweighting can successfully cover all the configurations whose cross section is enhanced by the change of parameters. On the other hand, in the presence of large reweighting factors, the statistical accuracy of the reweighted

[2] Notice however that weighted events are a necessity for precise Monte Carlo generators that include radiative loop corrections because the function to be sampled is not manifestly positive by an artifact of the loop expansion. Therefore, the weights can assume negative values and the events cannot be unweighted.

sample can be significantly reduced. These two issues can be monitored through some standard diagnostic tools, such as using the effective number of events $(\sum w_e(c))^2/\sum w_e(c)^2$ to estimate the statistical error, and ensuring that, for the values of the EFT coefficients used in the training process and in the analysis, the reweighting factors are not to large, i.e. $\sum w_e(c)/\sum w_e(0)$ does not differ too much from unity.

Event reweighting entails enormous practical advantages, which are well recognized in the literature [31–37]. The advantages are particularly striking for EFT applications. Therefore, event reweighting is well-developed in this context and is now fully automated. In particular, the MadGraph framework enables to generate EFT reweighted Monte Carlo samples also at the Next to Leading Order (NLO) accuracy in the QCD loop expansion [36, 38, 39].

The advantages of event reweighting are straightforwardly portable to the likelihood learning problem. The generation of the Monte Carlo data sets employed for training — rather than training itself — is a major or dominant component of the computational cost, especially at NLO. Event reweighting reduces this computational cost strongly because only one Monte Carlo data set is needed instead of the several data sets that are required to populate the c parameter space in the unweighted approach. In fact, unweighted event generation could become unfeasible in certain EFT applications where a large number of c parameters — denoted as Wilson coefficients in the EFT context — has to be considered because the number of required simulations grows rapidly with the dimensionality of c.

If the effect of the c parameters on the distribution is small, event reweighting is also beneficial for the accuracy of the predictions. More specifically, it improves the determination of the dependence on c of the predictions. In the unweighted approach, this dependence is extracted by comparing the SM prediction, for $c = 0$, with the one for $c \neq 0$ estimated from two independent sets of Monte Carlo events. The uncertainties on these predictions are ultimately determined by the resources that are invested in the generation, and in particular by the number of events in the Monte Carlo data set, which controls the Monte Carlo statistical uncertainties. The effect of the parameters is typically small in EFT analyses for realistic values of the Wilson coefficients. Very small uncertainties are thus needed in order to be sensitive to the dependence on c of the observables, which in turn requires very large Monte Carlo data sets, often beyond what is feasible in practice. In ref. [5] we solved this problem using unphysically large value of the Wilson coefficients for event generation and extrapolating down to realistic values exploiting the analytic knowledge of the (quadratic) dependence of the r ratio on c. This is a viable approach, which however requires a careful choice of the Wilson coefficient values. The choice is problem-specific and arguably difficult to automate.

Reweighted data enable a precise determination of the Wilson coefficients effects on observables, even if these effects are small, because the predictions are not affected by independent statistical uncertainties in the $c = 0$ and in the $c \neq 0$ data sets. Consider the simplest prediction of the cross section in a bin. We can determine the $c = 0$ cross section by summing the $w_\mathrm{e}(0)$ weights of the events that fall in the bin, the cross section for $c \neq 0$ by summing $w_\mathrm{e}(c)$, or even directly the difference between the two by summing up $w_\mathrm{e}(c) - w_\mathrm{e}(0)$ on the same events. In all cases, the finite Monte Carlo statistics will entail — if the weights are not vastly different — a relative uncertainty that is of the order of one over the square root of the number of events that fall in the bin. No matter how small the cross section difference is, a good relative accuracy on its prediction can be attained with manageable Monte Carlo statistics. We expect a similar advantage using reweighted data sets for the determination of the r ratio.

In this paper, we investigate the advantages of event reweighting through a direct comparison with ref. [5], by proceeding as follows. First, in section 2, we describe a straightforward adaptation to reweighted training sets of the methodology we developed in [5] for likelihood learning. Next, in section 3, we describe the application of the novel strategy to the case studies considered in ref. [5] and compare the performances. In section 2 we also introduce novel techniques — expanding ideas from ref. [5]—for the assessment of the quality of the r ratio reconstruction, which is crucial for hyper-parameters selection. Somewhat outside the main line of development of the paper, in section 4 we describe a frequentist proposal based on asymptotic formulas and the Asimov trick [40] for setting limits on the Wilson coefficients using the learned likelihood ratio, and outline its Bayesian interpretation. Our conclusions are in section 5.

2 Methodology

2.1 Learning from weights

A standard result in statistical learning theory — known as the likelihood-ratio trick — is that a continuous-output classifier trained to tell apart two data sets approximates the ratio between the probability distribution of the two training sets up to a given monotonic transformation. More precisely, the statement is that the classification function $f(x) \in (0,1)$ that minimizes the expectation value of the loss function — i.e., the loss of an infinite training set — is in one-to-one correspondence with the distribution ratio. The classification function that is actually obtained by training does not correspond to the exact distribution ratio because training sets are finite (estimation error), because the class of functions does not contain the exact ratio (approximation error), and because the optimization algorithm might not converge to the actual minimum of the loss. Monitoring and reducing these sources of error down to a satisfactory level is the universal goal of all practical applications of statistical learning including those of the present paper.

2.1.1 The simple classifier

All the methods [1–12] to extract the distribution ratio $r(x;c)$ in eq. (1.3) from Monte Carlo events are implementations of the likelihood-ratio trick. If the Monte Carlo data consists of a single set $\mathrm{S} = \{\mathrm{e}_i\}_{i=1}^{n(\mathrm{S})}$, where the dependence of the parameters of interest c is included by reweighting in the events $\mathrm{e} = (x_\mathrm{e},\, w_\mathrm{e}(c))$, as described in the Introduction, a straightforward adaptation of these ideas works as follows. Consider first the simpler task of learning the ratio, $r(x;\bar{c})$, at a fixed point $c = \bar{c}$ in the parameter space. This can be achieved with the loss function

$$\ell[f(\cdot)] = \sum_{\mathrm{e}\in\mathrm{S}} w_\mathrm{e}(\bar{c})\,[f(x_\mathrm{e})]^2 + \sum_{\mathrm{e}\in\mathrm{S}} w_\mathrm{e}(0)\,[f(x_\mathrm{e}) - 1]^2\,. \tag{2.1}$$

Notice that the two summations are performed on the same data set. The classification function f is thus evaluated on the same points in the two terms.

The events in S are by construction such that a weighted sum over them approaches, if the data set is large, the expectation value over the x variable multiplied by the total cross section. Given that the probability distribution function of x is equal to $d\sigma(x;c)/\sigma(c)$, the loss function for infinitely large S approaches

$$\ell[f(\cdot)] \to \int d\sigma(x;\bar{c})\,[f(x)]^2 + \int d\sigma(x;0)\,[f(x)-1]^2\,. \tag{2.2}$$

This functional attains its absolute minimum for

$$f(x) = \frac{1}{1 + d\sigma(x;\bar{c})/d\sigma(x;0)} = \frac{1}{1 + r(x;\bar{c})}\,, \tag{2.3}$$

which is in one-to-one correspondence with the ratio $r(x;\bar{c})$. By inverting the above equation we can thus turn the trained model, which minimizes the loss (2.1), into an estimate of $r(x;\bar{c})$.

It is interesting to compare eq. (2.1) with the loss function that one would employ instead in order to learn r from a Monte Carlo generator — either weighted or unweighted — that does not implement event reweighting. The expression would be very similar (see for instance eq. (6) of ref. [5]), but the two sums would be evaluated on two different event data sets, S_0 and S_1. The two sets are generated by independent runs, with the parameters c set to zero and to $\bar{c}$, respectively. They do not contain the same x_{e} points.

As described in the Introduction, event reweighting is in general beneficial for the accuracy of the predictions thanks to a reduced sensitivity to the Monte Carlo statistical fluctuations. Similar advantages are expected for the determination of the distribution ratio, most strikingly when $\bar{c}$ is small, enabling a more accurate determination of the small departures of $r(x;\bar{c})$ from one. A concrete verification and quantification of these advantages is postponed to section 3. In the rest of the present section, we describe our theoretical understanding of this behavior.

When $\bar{c}$ is small such that r is close to one, the optimal classification function (2.3) is close to $1/2$. In order to study this regime it is thus convenient to express $f(x) = 1/2 + \delta f(x)$, where $\delta f(x)$ is small in the optimal configuration (2.3), accounting for the small departures of r from one. The trained model configuration, which minimizes the loss function, is also characterized by a small δf. The question is whether this small learned δf is a good approximation of the optimal δf, producing an accurate determination of the departure of the r ratio from one. The loss function (2.1) reads

$$\begin{aligned}&\sum_{\mathrm{e}\in\mathrm{S}} w_{\mathrm{e}}(\bar{c})\,\delta f(x_{\mathrm{e}}) - \sum_{\mathrm{e}\in\mathrm{S}} w_{\mathrm{e}}(0)\,\delta f(x_{\mathrm{e}}) + \sum_{\mathrm{e}\in\mathrm{S}} w_{\mathrm{e}}(\bar{c})\,\delta f(x_{\mathrm{e}})^2 + \sum_{\mathrm{e}\in\mathrm{S}} w_{\mathrm{e}}(0)\,\delta f(x_{\mathrm{e}})^2\\ &= \sum_{\mathrm{e}\in\mathrm{S}} [w_{\mathrm{e}}(\bar{c}) - w_{\mathrm{e}}(0)]\,\delta f(x_{\mathrm{e}}) + \sum_{\mathrm{e}\in\mathrm{S}} [w_{\mathrm{e}}(\bar{c}) + w_{\mathrm{e}}(0)]\,\delta f(x_{\mathrm{e}})^2\,,\end{aligned} \tag{2.4}$$

up to an additive constant that is irrelevant for the minimization. If $\bar{c}$ is small, $w_{\mathrm{e}}(\bar{c})$ and $w_{\mathrm{e}}(0)$ are almost identical and the term in the loss function which is linear in δf is strongly suppressed. The suppression of the linear term in comparison with the quadratic one eventually makes δf small at the minimum of the loss function. Importantly enough, the linear term is small at each individual training point x_{e} at which δf is evaluated. This is outlined by the second line of eq. (2.4), where we collected under a single summation the linear and the quadratic terms. If the training sample is large, the summations provide good approximations of the δf and δf^2 terms of the loss function expectation (2.2). The relative accuracy of these approximations scales like one over the square root of the number of training points regardless of whether these terms are small or large. We will attain a similarly small relative accuracy in the determination of the optimal δf by the minimization of the loss function both if $\bar{c}$ is large or if it is small and the linear term is suppressed.

We now compare eq. (2.4) with the analogous expression that we would obtain instead when the two independent data sets S_0 and S_1 are used for training. We would get the terms on the first line of the equation, but they will be evaluated on the different data sets. In particular, the linear term will emerge from a cancellation between a summation over S_0 and one over S_1, with opposite sign. The relative statistical uncertainties of order one over the square root of the number of points will affect the two summations independently entailing, if $\bar{c}$ is small, a degradation of the accuracy in the reconstruction of the linear term. A good determination of δf would thus require large training samples and eventually become unfeasible for extremely small $\bar{c}$, preventing a determination of the departure of r from one. Using reweighted training data avoids this problem.

While presented in the case of quadratic loss, the considerations above hold for other choices of the loss function including the binary cross-entropy that is most often employed for classification. We tested the usage of the binary cross-entropy in our experiments, finding essentially identical results as for the quadratic loss. Since we did not encounter a case where it makes a difference, the impact of the choice of the loss function is not discussed further, and the quadratic loss is employed throughout this paper.

An interesting peculiarity of the training scheme based on reweighted data concerns the origin of overfitting. In regular training based on independent and unweighted data sets S_0 and S_1, the classifier function $f(x)$ receives, from the loss function minimization, a push to approach zero at the x_{e} points that belong to the S_0 set, and a push to approach one at the S_1 points. Since the x_{e} points in S_0 and in S_1 are distinct, this encourages the development of overfitted configurations where the model f wildly oscillates from zero to one in correspondence of individual training points. In the case of reweighted training data instead, there is only one set of x_{e} points where the loss function (2.1) is evaluated. Overfitting thus emerges from a different mechanism. We explained in the Introduction that the weights $w_{\mathrm{e}}(c) = w_{\mathrm{e}}(0)\, d\sigma(\xi;c)/d\sigma(\xi;0)$ are functions of latent variables, ξ, that are not in one-to-one correspondence with the observables x. Two events in the training set that are very near in the x space are typically far apart in the ξ space and hence their weights can be vastly different. The loss (2.1) pushes the classifier towards zero at the x_{e} point where $w_{\mathrm{e}}(\bar{c})$ is large, and towards one at the nearby point where it is small. This can produce overfitted configurations, by a mechanism that is slightly different than the one at work in regular classifier training. Avoiding overfitting in regular training requires regularization, which prevents the model to develop overly sharp features, and large training sets, which populate the x space densely. These same overfitting mitigation strategies turn out to be effective also for training with reweighted events.

2.1.2 The quadratic classifier for EFT

Learning the ratio between two specific distributions can be of practical interest. However, in most cases one needs instead the ratio $r(x;c)$ as a function of the parameters of interest c. The simple classifier described above can only learn $r(x;c)$ point-by-point in the c space, making extremely demanding or impossible to reconstruct the dependence on c, especially if there are several parameters of interest, $c = (c_1, \ldots, c_d)$. It is possible to overcome this limitation if the parametric dependence of $r(x;c)$ on c is known, by employing a *parametrized classifier* [5].

Parametrized classifiers are useful in cases where the distribution ratio can be parametrized in terms of a known function of c with x-dependent coefficient functions $\Gamma(x)$, namely if r reads

$$r(x;c) = \mathcal{P}\left(\Gamma(x);c\right) . \tag{2.5}$$

In this case, we know from eq. (2.3) the dependence on c of the optimal classification function $f = 1/(1+r)$. We can thus make an Ansatz for the functional form of the classification function

$$f(\gamma(x);c) = \frac{1}{1+\mathcal{P}\left(\gamma(x);c\right)} , \tag{2.6}$$

by employing a flexible class of functions — such as neural networks — to model the coefficient functions $\gamma(x)$. The absolute minimum of the loss function will be attained for $f = 1/(1+r)$, i.e. for $\gamma(x) = \Gamma(x)$. By training, we can thus produce γ functions that approximate the true Γ coefficient functions. By using the trained γ functions in eq. (2.5) in place of Γ, we eventually obtain an estimate of the ratio r.

The configuration $\gamma(x) = \Gamma(x)$ is a global minimum of the expectation value of any conceivable loss function, including eq. (2.1) with arbitrarily chosen $\bar{c}$. However, the minimum needs to be unique in order for the loss function minimization to determine all the different coefficient functions. The loss in eq. (2.1) possesses instead a family of degenerate global minima,[3] because it only contains information on the likelihood ratio at a single point $c = \bar{c}$. It is thus minimized by any of the many configurations that reproduce the likelihood ratio at that point. More points in the c space are needed for a unique determination of all the coefficient functions. We thus consider a set $\mathcal{C} = \{\bar{c}^{(1)}, \ldots, \bar{c}^{(\kappa)}\}$ of κ distinct points and we define another loss function

$$\ell[\gamma(\cdot)] = \sum_{\mathrm{e}\in\mathrm{S}} \sum_{\bar{c}\in\mathcal{C}} \left\{ w_{\mathrm{e}}(\bar{c}) \left[f(\gamma(x_{\mathrm{e}});\bar{c})\right]^2 + w_{\mathrm{e}}(0) \left[f(\gamma(x_{\mathrm{e}});\bar{c}) - 1\right]^2 \right\} . \tag{2.7}$$

This loss in the large-S limit has a unique global minimum for $\gamma(x) = \Gamma(x)$, provided the number κ of points in $\mathcal{C}$ is greater or equal than the number of coefficient functions to be determined.

The systematic exploration of the effect of heavy new physics described through the SM EFT is a prime target of the LHC, the HL-LHC, and future collider projects. In this context, the parameters of interest c are the Wilson coefficients of the new EFT interactions. In general, their effect on the differential cross sections $d\sigma(x;c)$ can be captured by a second-order polynomial. We could thus consider a parametrization

$$\mathcal{P}\left(\gamma(x);c\right) = 1 + \sum_{i=1}^{d} c_i \gamma_i(x) + \sum_{j\geq i}^{d} c_i c_j \gamma_{ij}(x) , \tag{2.8}$$

[3] This can be readily seen by rewriting the large-S limit of the loss, given by eq. (2.2), like in eq. (45) of ref. [5].

with a total of $d(d+3)/2$ γ functions, when d Wilson coefficients are present. The constant term equals one because $r(x;0) = 1$ by definition.

It is worth mentioning that the quadratic parametrization in eq. (2.8) does not capture all possible EFT effects. The EFT might affect parameters, like the mass or the width of SM particles, whose effect on the differential cross section is not polynomial. These contributions are often too small to be seen at the LHC. Still, they could be incorporated where needed by learning the dependence of the γ functions on masses or widths with a point-by-point approach. Including EFT contributions beyond the quadratic order, as they emerge from loop diagrams involving more than one EFT operator, would require generalizing eq. (2.8). However, these contributions are negligible. The quadratic parametrization accounts instead for relevant loops involving one EFT vertex and SM vertices, such as those associated with NLO QCD corrections.

A clear advantage [2] of employing the quadratic parametrization is that the number of distinct points in the c space to be considered for learning the complete c-dependent ratio $r(x;c)$ scales quadratically with the dimensionality of c. It scales instead exponentially in the point-by-point approach where the dependence on c is not imposed and needs to be reconstructed.

Another advantage [5] is that, by exploiting the parametrization, strategies can be devised for improving the quality of the likelihood reconstruction. We previously discussed that the ratio is difficult to learn accurately if the Wilson coefficients c are small, at least when unweighted training data are used. But we do not need to employ small values of c for the training of the parametrized classifier, in spite of the fact that the values that are relevant in the actual analysis of the data will eventually be small. The coefficient functions in the parametrization (2.8) can be learned by training with much larger c. If these functions are reconstructed with good relative accuracy, the departure from one of the r ratio is accurately reconstructed also for small c. The exact analytical knowledge of the dependence of $r(x;c)$ on c enables the extrapolation from large to small values. On the other hand, the c values used for training can not be arbitrarily large, otherwise the contribution of the quadratic polynomial terms in eq. (2.8) would dominate, the effect of the linear terms would be hidden and could not be learned. A proper selection of the c points in the set $\mathcal{C} = \{\bar{c}^{(1)}, \ldots, \bar{c}^{(\kappa)}\}$, used for training, is the main factor that controls the quality of the r ratio reconstruction. The selected c values should take into account the need of learning both the linear and the quadratic terms in all the different regions of the phase space. Since the absolute and relative magnitude of the two terms can vary radically in different regions of the phase space, several different $\bar{c}$ values are needed and must be included in the $\mathcal{C}$ set.

The strong sensitivity of the quality of the likelihood reconstruction to the choice of the training points requires case-by-case optimization, which is an obstruction to the automation of the methodology of ref. [5]. We will verify in section 3 that employing reweighted events reduces this sensitivity because it enables accurate learning of small effects as previously discussed.

An alternative [6] to our strategy, which also employs reweighted events, is to learn the linear and quadratic terms in separate training processes using suitable loss functions that contain individual terms of the polynomial expansion of the weight $w_{\rm e}(c)$. We made some attempts to adapt the strategy of [6] to our problem, without attaining satisfactory performances. An extensive comparison with other methods, including also the approach of ref. [10] possibly adapted to reweighted training data, is left to future work.

2.1.3 The EFT learning strategy

Several options could be considered for the practical implementation of the quadratic classifier approach. The simplest one would be to adopt the basic quadratic parametrization in eq. (2.8) using feed-forward neural networks, with output in $\mathbb{R}$, to model the $\gamma_i(x)$ and $\gamma_{ij}(x)$ coefficient functions. All the functions — there are $d(d+3)/2$ of them, for d Wilson coefficients — could be learned simultaneously in a single training using the loss in eq. (2.7) with $\kappa \geq d(d+3)/2$ training points in $\mathcal{C}$. A potential limitation of this scheme is that the basic parametrization (2.8) does not take into account that the r ratio, being the ratio of positive-defined physical cross sections, is itself positive. If the parametrization turns negative at some point in x and c at some stage of the training process, the classification function $f(x;c)$ in eq. (2.6) exits the $(0,1)$ interval and potentially diverges, for $\mathcal{P}\left(\gamma(x);c\right) = -1$. This risks producing training instabilities, especially if the cross-entropy loss was used in place of the quadratic loss. Furthermore, enforcing the physical constraint $r(x;c) > 0$ can be beneficial for the accuracy of the ratio reconstruction.

Following [5], we can enforce cross section positivity using the parametrization

$$\mathcal{P}\left(\lambda(x);c\right) = \sum_{I=1}^{d+1}\left[\sum_{J=1}^{d+1}\lambda_{IJ}(x)c_{J-1}\right]^2 , \tag{2.9}$$

where we defined $c_0 = 1$, and $\lambda(x)$ is an upper triangular real $(d+1)$-dimensional squared matrix with $\lambda_{11}(x) = 1$. This expression provides, for general $\lambda(x)$, the most general positive quadratic polynomial of c. Neural networks with output in $\mathbb{R}$ can be employed to model the non-trivial entries of the λ matrix.

The conceptually straightforward approach of learning all the coefficients functions in a single training, using the parametrization in eq. (2.9), suffers from practical limitations when the number d of Wilson coefficients is large. Typically available GPUs have limited memory. They can hardly accommodate all the gradients that have to be stored for the training of more than around 10 reasonably complex neural network models using the large number of training points required for an accurate learning of the coefficient functions. This prevents using GPUs for more than 2 or 3 Wilson coefficients, with a dramatic impact on the training execution time.[4] A different scheme is needed, in which the different polynomial terms of the basic parametrization of eq. (2.8) are learned in separate trainings. These individual training stages could be run in parallel, if several GPUs are available, or sequentially on a single GPU still entailing a strong improvement of the execution time in comparison with CPU training.

A scheme that is suitable for parallelization works as follows. Since the polynomial is quadratic, all its coefficients can be extracted by considering configurations where only two Wilson coefficients — in all possible pairings — are turned on, while the others are set to zero. Let us then start from the case of a 2-dimensional coefficient vector $c = (c_1, c_2)$, for which we can employ the manifestly positive parametrization in eq. (2.9), with $d = 2$. For reasons that will become momentarily clear, we do not employ neural networks to model the

[4] Using mini-batches can circumvent GPU memory limitations, but slows down training. Notice that the mini-batch gradients need to be accumulated and the weight update step taking only after the whole training data set is processes. This is because an accurate determination of the gradients pointing towards the true minimum of the loss function is needed for an accurate learning.

entries of the λ matrix directly. We instead express the upper triangular matrix λ as

$$\lambda(x) = \begin{pmatrix} 1 & \rho_1(x)\sin\theta_{11}(x) & \rho_2(x)\sin\theta_{22}(x) \\ 0 & \rho_1(x)\cos\theta_{11}(x) & \rho_2(x)\cos\theta_{22}(x)\sin\theta_{12}(x) \\ 0 & 0 & \rho_2(x)\cos\theta_{22}(x)\cos\theta_{12}(x) \end{pmatrix}\,, \tag{2.10}$$

using polar and spherical coordinates for the second and third columns, respectively. With this expression, the parametrization (2.9) of the distribution ratio becomes

$$\begin{aligned}\mathcal{P}(\rho_1,\rho_2,\theta_{11},\theta_{22},\theta_{12};c) = 1 &+ 2\,c_1\rho_1\,\sin\theta_{11} + c_1^2\rho_1^2 + 2\,c_2\rho_2\,\sin\theta_{22} + c_2^2\rho_2^2 \\ &+ 2\,c_1c_2\rho_1\rho_2\left(\cos\theta_{11}\cos\theta_{22}\sin\theta_{12} + \sin\theta_{11}\sin\theta_{22}\right)\,. \end{aligned}\tag{2.11}$$

We employ neural networks to parametrize the two radial functions $\rho_{1,2}$ and the three angular functions θ_{11}, θ_{22} and θ_{12}. We consider networks with unconstrained outputs spanning the whole real axis, and we ignore the periodicity of the angular functions and the positivity of the radial functions. This choice does not invalidate the generality and the positivity of our parametrization. It merely makes it redundant, which is not a limitation: during training, the networks will pick up one of the equivalent configurations that correspond to the correct distribution ratio. No training instability will emerge because each equivalent configuration is a separate global minimum of the loss function.

If only $d = 2$ Wilson coefficients are present, the GPU memory is probably sufficient to store all the gradients and the 5 neural networks $\rho_{1,2}$ and $\theta_{11,22,12}$ could be learned in a single training. Consider however the following alternative, which is suited for generalization to the case $d > 2$. In eq. (2.11), the c_1 and c_1^2 terms are fully determined by the ρ_1 and the θ_{11} networks. These networks can thus be learned separately from the others by training data involving only c_1, with vanishing c_2. Similarly, we can learn ρ_2 and θ_{22} by training with c_2 only. Training data where both c_1 and c_2 are non-vanishing are only needed in order to learn θ_{12}, which in turn gives access to the mixed polynomial term c_1c_2. Learning θ_{12} requires the knowledge of $\rho_{1,2}$ and of $\theta_{11,22}$, thus the determination of θ_{12} can not proceed in parallel with the determination of the other networks. However, while training θ_{12}, the $\rho_{1,2}$ and $\theta_{11,22}$ networks can be kept frozen to their previously-learned configurations and they are not optimized. The only gradients to be stored in the GPU memory are those of the θ_{12} network parameters.

If $d > 2$ Wilson coefficients are present, one can first run d separate trainings with a single non-vanishing Wilson coefficient, c_i, using a parametrization

$$\mathcal{P}(\rho_i,\theta_{ii};c_i) = 1 + 2\,c_i\rho_i\,\sin\theta_{ii} + c_i^2\rho_i^2\,, \tag{2.12}$$

which is the one-dimensional version of eq. (2.11). These training stages can operate in parallel and they give access to the c_i and c_i^2 terms of the polynomial, for all $i = 1,\ldots,d$. Next, we turn on a pair (c_i, c_j) of Wilson coefficients and consider a 2-dimensional parametrization like the one of eq. (2.11), with networks ρ_i, ρ_j, θ_{ii}, θ_{jj} and θ_{ij}. The ρ_i, ρ_j, θ_{ii} and θ_{jj} networks are those of eq. (2.12), and they were previously learned in the one-dimensional trainings. The remaining network, θ_{ij}, can be trained keeping the others fixed, as previously explained. This gives access to the c_ic_j mixed term. Extracting all the mixed terms requires

$d(d-1)/2$ separate trainings — corresponding to the distinct pairings of Wilson coefficient — that can run in parallel. Finally, once all the polynomial coefficients are known, they can be pulled together in eq. (2.8) providing the full knowledge of the distribution ratio everywhere in the Wilson coefficients space.

For the case studies of the present paper, with $d = 2$, the distribution ratio parametrized by eq. (2.11) can be learned in a single training and the protocol described above is not needed. However, it will be essential in order to deal with the large number of Wilson coefficients that are required, as emphasized in [10], for global EFT fits. Our methodology offers an ideal solution to this problem. It enables parallelization or serial execution on GPUs with limited memory while, at the same time, rigorously enforcing the physical constraint of distribution ratio positivity at all training stages. The only potential limitation, in comparison with directly learning all the coefficient functions of the general manifestly positive parametrization in eq. (2.9), is that our strategy based on learning the individual terms of the basic polynomial parametrization (2.8) does not produce a distribution ratio that is necessarily positive when more than 2 coefficients are non-vanishing. It is unclear whether negative ratios will be ever encountered. Their occurrence for the small values of the Wilson coefficients we will be interested in probing would signal overfitting because the true effect of the EFT operators should typically produce a small correction to the SM cross section that can hardly cause a strong departure of the ratio from unity.

2.2 Performance metrics

Learning the distribution ratio with the strategy described in the previous section is not very different from training a classifier. It requires choosing a model — feed-forward neural networks, in our case — and selecting its parameters as well as training hyper-parameters like the training sample size, the number of epochs and the learning rate plus, if needed, explicit regularization parameters. The loss function (2.7) also features the number and the values of the Wilson coefficients in the $\mathcal{C}$ set, $\vec{c}^{(1)}, \dots, \vec{c}^{(\kappa)}$, as additional hyper-parameters. However, when training a classifier for regular classification purposes one operates hyper-parameter selection based on figures of merit that quantify the performances of the trained classification function. Standard performance metrics are the accuracy or the AUC. Our scope is not classification, we thus need to define different performance metrics. This is the goal of the present section.[5]

Our goal is to extract a good approximation of the distribution ratio $r(x; c)$. Denoting as $\hat{r}(x; c)$ the reconstructed ratio, the most straightforward performance metrics to assess the quality of our results should thus measure some sort of distance between the true ratio $r(x; c)$ and $\hat{r}(x; c)$. Even if the true ratio is of course unknown, valid notions of distance can be constructed as follows. The true distribution ratio relates the SM differential cross section, $d\sigma(x; 0)$, to the c-dependent cross section, $d\sigma(x; c)$, by

$$d\sigma(x; c) = r(x; c)\, d\sigma(x; 0)\,. \tag{2.13}$$

We can thus define an approximate cross section

$$d\hat{\sigma}(x; c) = \hat{r}(x; c)\, d\sigma(x; 0)\,, \tag{2.14}$$

[5] Most of what follows is a refinement of ideas we first presented in ref. [5].

by employing $\hat{r}$ in place of r. Comparing $d\sigma(x;c)$ with $d\hat{\sigma}(x;c)$ measures the distance between r and $\hat{r}$. We do not have access to the differential cross section, but we can compute the cross section $\Delta\sigma(c)$ integrated in some bin using Monte Carlo events. If the events are reweighted, the cross section is the sum of the weights of the events that fall into the bin

$$\Delta\sigma(c) = \sum_{x_{\rm e}\in{\rm bin}} w_{\rm e}(c)\,. \tag{2.15}$$

This quantity can be compared with the integral of $d\hat{\sigma}(x;c)$ in the bin

$$\Delta\hat{\sigma}(c) = \sum_{x_{\rm e}\in{\rm bin}} w_{\rm e}(0)\,\hat{r}(x_{\rm e};c)\,. \tag{2.16}$$

By performing this comparison for relevant binned univariate marginals, we can visualize the quality of the distribution ratio reconstruction. Results are presented in section 3. See for instance figure 3.

An alternative measure of the distance between $\hat{r}$ and r, which does not rely on the arbitrary selection of marginals, can be defined by exploiting a certain characteristic property of the true distribution ratio $r(x;c)$, seen as a one-dimensional variable depending on x, for fixed c. We actually work with the logarithm of this variable, τ_c, and with its reconstruction, $\hat{\tau}_c$, namely

$$\tau_c(x) = \log r(x;c)\,, \qquad \hat{\tau}_c(x) = \log\hat{r}(x;c)\,. \tag{2.17}$$

The differential cross-section for the τ_c variable is the integral of $d\hat{\sigma}(x;c)$ on slices of fixed $r(x;c)$. Hence, using eq. (2.13), we have

$$d\sigma(\tau_c;c) = e^{\tau_c} d\sigma(\tau_c;0) \;\;\Rightarrow\;\; \log\left[d\sigma(\tau_c;c)/d\sigma(\tau_c;0)\right] = \tau_c\,. \tag{2.18}$$

The reconstructed log-ratio variable, $\hat{\tau}_c$, does not obey eq. (2.18) because it is not the logarithm of the true distribution ratio, but of the reconstructed one. We can thus quantify the quality of the approximation by studying the validity of eq. (2.18) for the differential distributions of the $\hat{\tau}_c$ variable: $d\sigma(\hat{\tau}_c;c)$ and $d\sigma(\hat{\tau}_c;0)$. We proceed by binning $\hat{\tau}_c$ and computing, for each bin with boundaries (τ_-,τ_+), the integrated cross section

$$\Delta\sigma(c) = \sum_{\tau_c(x_{\rm e})\in(\tau_-,\tau_+)} w_{\rm e}(c)\,. \tag{2.19}$$

For each bin we then compute

$$\mathcal{T}_c = \log\Big[\Delta\sigma(c)/\Delta\sigma(0)\Big]\,. \tag{2.20}$$

If eq. (2.18) is approximately verified for the $d\sigma(\hat{\tau}_c;c)$ and $d\sigma(\hat{\tau}_c;0)$ cross sections, and if the bin is sufficiently narrow, $\mathcal{T}_c$ approximately follows a straight line, $\mathcal{T}_c \simeq \tau_{\rm avg}$ with $\tau_{\rm avg} = (\tau_+ + \tau_-)/2$. By plotting $\mathcal{T}_c$ in the different bins we can thus visualize the validity of eq. (2.18). One example is shown, for instance, on the left panel of figure 4.

A further refinement exploits that if $\hat{\tau}_c$ was equal to the true τ_c, the cross section $\Delta\sigma(c)$ in the bin $\hat{\tau}_c(x) \in (\tau_-,\tau_+)$ would be bounded by

$$e^{\tau_-}\Delta\sigma(0) < \Delta\sigma(c) = \int_{\tau_-}^{\tau_+} d\sigma(\hat{\tau};c) < e^{\tau_+}\Delta\sigma(0)\,, \tag{2.21}$$

using $\hat{\tau}(x) = \tau(x)$, and eq. (2.18). Therefore, if $\hat{\tau}_c$ was equal to τ_c, $\mathcal{T}_c$ would be bounded as

$$\tau_- < \mathcal{T}_c < \tau_+ \quad \rightarrow \quad |\mathcal{T}_c - \tau_{\text{avg}}| < \frac{\tau_+ - \tau_-}{2}\,. \tag{2.22}$$

Verifying if $\mathcal{T}_c$ sits in this window — see for instance the right panel of figure 4—gives an indication of the validity of eq. (2.18) for the reconstructed ratio that is more quantitative than checking qualitatively the relation $\mathcal{T}_c \simeq \tau_{\text{avg}}$.

2.2.1 The Neyman-Pearson *p*-value: definition

The previously-described strategies provide an important assessment of the quality of the r ratio reconstruction. We have found them effective in practice to discriminate between different trained models and therefore helpful for hyper-parameters selection. However, they do not constitute our prime performance metric, because of two reasons. First, because it is difficult to condense the information they provide into a single quality indicator. Second, because they do not offer an objective criterion to quantify the improvement obtained by a certain configuration in comparison with another one. One can improve indefinitely the quality of the reconstruction by employing more computational resources i.e., typically, by using more training data and bigger neural networks. In order to balance performances against computational resources, we must be able to judge the significance of the improvement attained by a given configuration, relative for instance to a configuration with smaller networks and less data.

Our prime performance indicator is still sensitive to the distance between $r(x;c)$ and $\hat{r}(x;c)$, but less directly than the other ones. Ultimately, we seek access to r in order to model the likelihood log-ratio λ, in eq. (1.2), of a given collider experiment. In turn, its knowledge would enable us to extract maximal — i.e., optimal — statistical information on the parameters of interest from the experimental data. It is thus natural to measure the quality of the reconstructed ratio $\hat{r}$ in terms of its statistical performances, if used in place of r in eq. (1.2), on the collider experiment under examination. We thus define the reconstructed likelihood log-ratio

$$\hat{\lambda}(c;\mathcal{D}) = N(0) - N(c) + \sum_{x\in\mathcal{D}} \log \hat{r}(x;c) = N(0) - N(c) + \sum_{x\in\mathcal{D}} \hat{\tau}_c(x)\,, \tag{2.23}$$

where, as explained in the Introduction, $N(c) = L\,\sigma(c)$, with L the integrated luminosity of the experiment. The total cross section, $\sigma(c)$, is taken from the Monte Carlo with negligible error.

Several different statistical analyses could be potentially performed by employing the reconstructed likelihood log-ratio $\hat{\lambda}$ (2.23), ranging from setting a limit on the allowed size of the c parameters, excluding the SM point $c = 0$, or measuring the parameters. Classical or Bayesian methodologies could be employed. Without committing to any of these options for the analysis of the actual data, here we pick up one statistical analysis that could be performed — in line of principle — using $\hat{\lambda}$ and that is endowed with a sharp guarantee of statistical optimality if the learned likelihood log-ratio $\hat{\lambda}$ was exactly equal to the true log-ratio λ. The statistical performances of this analysis improve as $\hat{\lambda}$ approaches λ, providing another way to quantify the agreement between $\hat{r}$ and r. The performances saturate when the agreement between $\hat{r}$ and r is sufficient, and do not improve indefinitely. When saturation occurs and the performance gain stops being significant, it means that the reconstructed $\hat{\lambda}$ contains the same information as λ on the parameters of interest and no further improvement of the ratio reconstruction is needed.

The Neyman-Pearson Lemma [41] guarantees the optimality of employing the true likelihood ratio in order to discriminate between two different values c_0 and c_1 of the parameters of interest. The one considered in the Lemma is a test of hypothesis that could in principle be used to exclude the existence of EFT interaction operators with a specific value c of the Wilson coefficients. The performances of such exclusion analysis would be quantified — prior to the experiment — in terms of its expected sensitivity to non-vanishing c under the SM hypothesis that no EFT interaction exists and thus c is equal to 0. In the standard Neyman-Pearson notation, we should thus identify $c_0 = c$ and $c_1 = 0$ as the null and the alternative hypothesis, respectively.

A generic test of hypothesis works by defining a "test statistic" variable, $t(\mathcal{D})$, which depends collectively on the whole set of data $\mathcal{D} = \{x_i\}_{i=1}^{\mathcal{N}}$ that are collected in the experiment. Any quantity could be employed as a test statistic, in line of principle. However, a meaningful test statistic is one that is typically small when the null hypothesis H_0 ($c \neq 0$, in our case) is true, and large if instead the alternative hypothesis H_1 (i.e., $c = 0$) is true. Observing on the data a value of t that is way larger than the typical values of t attained in the presence of the EFT interactions disfavors the presence of the new interactions. A statistical notion of typicality is provided by the p-value

$$p_c(t) = \int_t^{\infty} dt' \mathrm{pdf}(t'|c)\,. \tag{2.24}$$

The p-value relates the observed value of t to the probability that an even larger value is observed when the EFT interactions are present in the data distribution. Small p_c signals that EFT interactions with Wilson coefficient c are unlikely to be present.

An *efficient* [41] hypothesis test for c exclusion is one that excludes with high confidence, i.e. with low p-value, if the SM hypothesis $c = 0$ is true. The metric that quantifies the test efficiency thus considers the typical value of $t(\mathcal{D})$ in the SM hypothesis, and the corresponding p-value (2.24). We use the median of t and, since $p_c(t)$ is monotonic in t, define the median p-value

$$p(c) = \mathrm{Median}\big[p_c(t)|c = 0\big]\,. \tag{2.25}$$

The median p-value is the figure of merit that quantifies the efficiency of hypothesis tests. Lower $p(c)$ indicates better performances.

Any choice of the test statistic variable $t(\mathcal{D})$ defines a valid test, whose efficiency is evaluated by the median p-value (2.25). However, the Lemma [41] identifies the most efficient hypothesis test as the one that employs as test statistic minus the logarithm of the ratio between the likelihood in the null and in the alternative hypotheses, times a conventional factor of two

$$t_c(\mathcal{D}) = -2\,\lambda(c;\mathcal{D}) = 2\Big[N(c) - N(0) - \sum_{x\in\mathcal{D}} \tau_c(x)\Big]\,. \tag{2.26}$$

The Lemma guarantees that the test based on $t_c(\mathcal{D})$ (or a monotonic function of it) has the lowest possible median p-value. Any other variable has inferior performances, namely a larger median p-value. In particular, this means that the performances of the reconstructed likelihood log-ratio

$$\hat{t}_c(\mathcal{D}) = -2\,\hat{\lambda}(c;\mathcal{D}) = 2\Big[N(c) - N(0) - \sum_{x\in\mathcal{D}} \hat{\tau}_c(x)\Big]\,, \tag{2.27}$$

are not optimal, but they approach the optimum as the reconstructed $\hat{\lambda}$ approaches the true λ. We can thus employ the median p-value of the $\hat{t}_c(\mathcal{D})$ test statistic as a metric to evaluate the performances of the ratio reconstruction. We denote this quantity as $\hat{p}(c)$.

Extensive use of $\hat{p}(c)$ is made in section 3 to compare different models and eventually select the hyper-parameters. Various types of performance studies are conducted. For instance, the evolution of $\hat{p}(c)$ during the training of a certain model is displayed on the right panel of figure 2. The value of c considered in the figure has been chosen such that $\hat{p}(c) \approx$ few%, close to the threshold of 5% that is often conventionally considered to set an exclusion limit. This is because we want to probe the quality of the ratio reconstruction when c is close to the values that will be eventually relevant for the statistical analysis of the data. Alternatively, we can identify the region where $\hat{p}(c)$ is exactly equal to 5% and draw exclusion contours, as in figure 5. Figure 6 exemplifies instead the plots we made for hyper-parameter selection. It shows, among other things, the saturation of $\hat{p}(c)$ for increasingly complex networks and more training data. Section 3 describes these results extensively.

It should be noted that observing the saturation of $\hat{p}(c)$ does not guarantee that optimal performances are attained: the p-value evolution might decrease very slowly towards its (unknown) global minimum. However, a very slow decrease suggests that a significant performance improvement, if any, would require radically larger training data sets and neural networks, beyond what is feasible in practice. We can thus stop improving the quality of the ratio reconstruction as soon as saturation is observed. The same criterion is routinary adopted for the training of regular classifiers. Also in that context, the model improvement is stopped when the performances saturate, without guarantee that optimal performances have been attained.

2.2.2 The Neyman-Pearson p-value: calculation

The determination of the median p-value $\hat{p}(c)$ is conceptually straightforward, but numerically cumbersome. The direct approach is to generate artificial instances of the experimental data set $\mathcal{D} = \{x_i\}_{i=1}^{\mathcal{N}}$ that follow the H_0 hypothesis with non-vanishing c, and data that follow H_1, with $c = 0$. These pseudo-data — called toy data — are build by first drawing the total number $\mathcal{N}$ from a Poisson distribution with expected $N(c)$ or $N(0)$, as in the hypothesis under examination, and next extracting $\mathcal{N}$ instances of the x variable, following the appropriate distribution, from Monte Carlo data. Toy data with $c \neq 0$ serve to determine empirically the distribution of the $\hat{t}_c$ variable (2.27) and in turn the $p_c(t)$ function. From the $c = 0$ toys one computes the median and $\hat{p}(c)$ (2.24). This procedure is feasible, but too slow to repeatedly compute $\hat{p}(c)$ for performance evaluation and hyper-parameters scan. Two alternative approaches are described below.

The first approach [5] is to model analytically the probability density functions of $\hat{t}_c$ under the two hypotheses $c \neq 0$ and $c = 0$. The former probability function determines $p_c(t)$ (2.27), while the knowledge of the latter one enables to compute the median in eq. (2.24). Such analytical modeling is possible because $\hat{t}_c$ (2.27) is trivially related to the sum over the data set of the variable $\hat{\tau}_c(x)$. Since the data set is large, the Central Limit theorem ensures that the distribution of $\hat{t}_c$ is approximately Gaussian. Departures from Gaussianity can be taken into account by modeling the $\hat{t}_c$ distribution with a skew-normal distribution [5]. Its free parameters, namely the mean, variance and skewness are related to the moments of $\hat{\tau}_c(x)$ by

$$\mu(\hat{t}_c) = 2\big[N(c) - N(0) - N\,\langle\hat{\tau}_c\rangle\big]\,, \qquad \sigma^2(\hat{t}_c) = 4\,N\,\langle\hat{\tau}_c^2\rangle\,, \qquad \mu_3(\hat{t}_c) = \frac{1}{\sqrt{N}}\,\frac{\langle\hat{\tau}_c^3\rangle}{\langle\hat{\tau}_c^2\rangle^{3/2}}\,. \tag{2.28}$$

In the equation, N denotes the expected number of events when either $c \neq 0$ or $c = 0$. The expectation value, denoted as $\langle\ldots\rangle$, is taken either under the $c \neq 0$ or the $c = 0$ hypotheses in order to determine the two distributions of $\hat{t}_c(\mathcal{D})$.

This semi-analytical approach to the calculation of $\hat{p}(c)$ is extremely fast, especially when using reweighted events that enable the determination of the averages in eq. (2.28) using the same Monte Carlo sample for any value of c. Implemented on a GPU, it can be run during training enabling online monitoring of the performances. It provides results that are normally accurate, as one can verify by comparing with the empirical evaluation of $\hat{p}(c)$ based on toy experiments. In some rare cases, however, the semi-analytical estimate of $\hat{p}(c)$ fails due to the failure of the quasi-Gaussian approximation for the distribution of $\hat{t}_c$. This typically occurs in networks that slightly overfit producing overly sharp peaks in $\hat{\tau}_c(x)$. The contribution of the peaks to $\hat{t}_c$ emerges from a small region in the x space, where few events are present. This violates the Central Limit theorem even if the total number $\mathcal{N}$ of events is large. When the Central Limit theorem violation occurs, the estimate of $\hat{p}(c)$ can be either much larger or much smaller than the true p-value.

A determination of $\hat{p}(c)$ that is more robust, but computationally more demanding, is obtained through quantile binning for the SM hypothesis as follows. We consider the discretization of the variable $\hat{\tau}_c(x)$ in a large number $n_{\rm bin}$ of non-overlapping bins. Namely, we approximate $\hat{\tau}_c(x)$ with a piecewise constant function:

$$\hat{\tau}_c(x) \;\simeq\; \{\tau_c[\mathrm{b}]\,,\ \text{for } x\ \text{ s.t. } \hat{\tau}_c(x) \in (\tau_{\mathrm{b}-1},\, \tau_{\mathrm{b}})\}\ , \tag{2.29}$$

where $\tau_0 = -\infty$ and $\tau_{n_{\rm bin}} = +\infty$ in order to cover the whole space. The approximately constant value of $\hat{\tau}_c$ in each bin, $\tau_c[\mathrm{b}]$, equals the central point of the bin apart from the extremes, where we set $\tau_c[1] = \tau_1$ and $\tau_c[n_{\rm bin}] = \tau_{n_{\rm bin}-1}$. Using this approximation, eq. (2.27) becomes

$$\hat{t}_c(\mathcal{D}) \simeq 2\left[N(c) - N(0) - \sum_{\mathrm{b}} \mathcal{N}_{\mathrm{b}}\tau_c[\mathrm{b}]\right], \tag{2.30}$$

where $\mathcal{N}_{\rm b}$ is the number of points in $\mathcal{D}$ that fall in the bin. $\mathcal{N}_{\rm b}$ follows a Poisson distribution with expected $L\,\Delta\sigma_{\rm b}(c)$, or $L\,\Delta\sigma_{\rm b}(0)$, with L the luminosity of the experiment. Highly optimized computer packages exist to generate Poisson-distributed numbers. We can thus efficiently determine $\hat{p}(c)$ empirically by generating toy data in the $c \neq 0$ and in the $c = 0$ hypotheses.

The binned determination of $\hat{p}(c)$ relies on the choice of a binning strategy, and of the number of bins $n_{\rm bin}$. Binning is performed by ensuring that the cross sections of all bins are equal under the SM hypothesis $c = 0$. The number of bins must be large enough to ensure that the binned test statistic in eq. (2.30) is a good enough approximation of un-binned $\hat{t}_c$ in eq. (2.27). At the same time, using too many bins reduces the accuracy of the determination of $\Delta\sigma_{\mathrm{b}}$ due to the finite Monte Carlo statistics. For the studies performed in section 3, we found that $n_{\mathrm{bin}} = 1000$ is a good compromise.

3 Performance studies

We illustrate the performances of our methodology on the same case study considered in ref. [5]. This will enable a direct assessment of the advantages of training with reweighted Monte Carlo data, rather than populating the EFT Wilson coefficients parameter space by independent data sets as done in [5]. We consider the production of a Z and of a W boson at the 14 TeV LHC, and their decay to leptons. We restrict our analysis to the high energy regime, with a cut of 300 GeV on the transverse momentum of the bosons. We study the sensitivity of this process, with the full integrated luminosity $L = 3\,\mathrm{ab}^{-1}$ of the HL-LHC, to two specific dimension-six EFT interactions

$$\mathcal{O}_\varphi = G_\varphi \left(\overline{Q}_L \sigma^a \gamma^\mu Q_L\right) (iH^\dagger \overleftrightarrow{D}_\mu H)\,, \qquad \mathcal{O}_W = G_W \varepsilon_{abc} W_\mu^{a\,\nu} W_\nu^{b\,\rho} W_\rho^{c\,\mu}\,. \tag{3.1}$$

See appendix A for an extensive description of the process and of the two Monte Carlo generators, namely the ideal and NLO generators, that we employ to model the distributions and the effect of the Wilson coefficients $c = (G_\varphi,\, G_W)$.

We do not aim at fully realistic sensitivity projections, nor at a comparative assessment of the ZW process sensitivity to the EFT operators at hand and its role in a global fit. Nevertheless, it is worth emphasizing [42–49] that the ZW process at high energy is a promising probe of the operators $\mathcal{O}_\varphi$ and $\mathcal{O}_W$, because these operators produce growing-with-energy effects in the ZW scattering amplitudes as displayed in eq. (A.7). The unique ability of the LHC to probe the operators in the high energy regime, where their effects are enhanced, can boost the sensitivity to their Wilson coefficients well beyond the current bounds from measurements performed at lower energy. Furthermore, the three leptons from the ZW decay define a sufficiently complex final state to expect a gain in sensitivity from an unbinned multivariate analysis in comparison with a more standard approach based on binned measurements of one- or two-dimensional distributions. A sensitivity gain by a factor more than 2 was demonstrated in [5] for the G_W Wilson coefficient, while the gain on G_φ is more moderate. The different behavior of the two operators is due — see [5, 46]—to the different role that is played by the kinematical variables that describe the decay of the vector bosons: their measurement is essential in order to access the growing-with-energy linear term in G_W, while the growing-with-energy linear G_φ term is present already in the differential di-boson cross section integrated over the decay angles. A simple binned analysis that does not exploit the distribution of the decay angles fully is thus nearly optimal for G_φ and vastly sub-optimal in the case of G_W. At the technical level, this difference makes the G_W contribution to the distribution ratio $r(x; c)$ a much more intricate function of the kinematical variables x than the G_φ contribution. Learning the G_W contribution is thus a harder problem than learning the G_φ contribution.

We consider the problem of learning the distribution ratio based on two different Monte Carlo generators: the ideal and the NLO generators. The ideal generator offers a simplified description of the process. It is based on approximations that are not sufficiently accurate for the actual analysis of the data, but enable a simple analytical determination of the true distribution ratio $r(x;c)$. Applying our methodology to the ideal setup provides a validation of the performances against ground-truth knowledge, on a learning problem of realistic complexity. The NLO generator provides instead an accurate state-of-the-art description of the ZW process. Studying the problem with the NLO generator validates our methodology in a realistic setup and tests its ability to deal with events with negative weight, whose presence is unavoidable for event generation beyond the tree level. Appendix A provides an extensive technical description of the ideal and NLO Monte Carlo generators and of how they are employed to produce reweighted data sets.

The implementation of our learning strategy (defined in section 2.1) and the study of its performances (based on the metrics of section 2.2) on ideal and on NLO data is described in the next two sections in turn. The direct comparison between the reconstructed and true distribution ratios will be also employed as a performance indicator in the case of ideal data.

All our models are implemented in `PyTorch` version 1.11.0 [50] and CUDA 11.3. All trainings were performed on NVIDIA A30 GPU and employing the Adam optimization algorithm [51].

3.1 Ideal data

Reweighted ideal Monte Carlo data sets $\mathrm{S} = \{\mathrm{e}_i\}_{i=1}^{n(\mathrm{S})}$, with $\mathrm{e} = (x_\mathrm{e},\, w_\mathrm{e}(c))$, are sampled from the ideal Monte Carlo generator — described in appendix A—implemented in a dedicated code. A total of 20 million events have been generated to produce the results that follow. Ten million are used for testing purposes, namely for the evaluation of the performance metrics introduced in section 2.2. Training is performed with $n(\mathrm{S}) = 3\mathrm{M}$ if not specified otherwise. An independent sample of the same size is employed for validation during training. The weights $w_\mathrm{e}(c)$ are computed by reweighting in the latent space as in eq. (A.11).

Each event is characterized by seven independent observable variables x, listed in eq. (A.10). It facilitates the learning task to pre-process the input by performing change of variables that avoid overly sharp one-dimensional marginal distributions, and by introducing redundancies. We pre-process with the transformation

$$x \to \left\{ \log[s/\mathrm{GeV}^2],\, \Theta,\, \theta_Z,\, \theta_W,\, \log[p_T/\mathrm{GeV}],\, Q,\, \sin\varphi_Z,\, \sin\varphi_W,\, \cos\varphi_Z,\, \cos\varphi_W \right\}. \tag{3.2}$$

The neural networks we employ for our analysis thus receive a total of 10 features as input, ordered as in the above equation. A normalization layer in the network shifts and scales each variable to have zero mean and unit variance on the training sample. Our pre-processing (3.2) follows relatively standard practice: the steeply-falling distribution of the center of mass energy squared, s, is smoothed out by taking the logarithm. The 2π periodicity of the azimuthal angles is enforced explicitly by giving the sine and the cosine, rather than the angle itself, as input to the network. The variable $p_T = \sqrt{s}/2\,\sin\Theta$ can be more useful to the network than s and Θ to model the effect of the EFT operators in some kinematical regimes.

3.1.1 The simple classifier

We start from the problem of learning the distribution ratio $r(x;\bar{c})$ at a specific point $c = \bar{c}$ of the Wilson coefficients parameter space. As explained in section 2.1.1, this is achieved using the loss function in eq. (2.1) to train a classifier $f(x) \in (0,1)$. We consider the classification function

$$f(x) = \frac{1}{1+e^{\rho(x)}}\,, \tag{3.3}$$

where $\rho(x)$ is a feed-forward neural network with real output. By comparing with eq. (2.3) we see that, after training, the neural network provides an approximation, $\hat{r}(x,\bar{c})$, of the true distribution ratio $r(x,\bar{c})$. More precisely

$$\log \hat{r}(x,\bar{c}) = \hat{\tau}_{\bar{c}}(x) = \rho(x)\,. \tag{3.4}$$

The value of $\bar{c}$ chosen for illustration has $G_\varphi = 0$ and $G_W = 10^{-2}\,\text{TeV}^{-2}$. Learning the distribution ratio with the simple classifier for such a small value of G_W was found to be possible in ref. [5], but only modest accuracy could be attained. Better performances are expected using reweighted training data for the reasons explained in section 2.1.1.

We use 3M training points, a ρ neural network with $(10,64,64,1)$ architecture — namely, two hidden layers with 64 neurons each — and sigmoid activation functions. Pre-processing is performed as in eq. (3.2). Training employs an initial learning rate of $3\cdot10^{-3}$ for the first 20k epochs, after which the initial learning rate parameter is lowered to 10^{-3} without re-initializing the optimizer. This 2-step training scheme with decreasing initial learning rate is found to be convenient in general: the first stage reduces the loss quickly, but the precise optimization with a lower rate performed at the second stage is required in order to attain a deeper minimum of the validation loss. Each training step is performed computing the gradients of the network parameters on the whole training set. Updating the networks with mini-batches is found, consistently with ref. [5], to degrade the accuracy of the distribution ratio reconstruction strongly. A validation loss is computed on 3M independent Monte Carlo data. Training is stopped at the minimum of the validation loss, which is reached in this case after around 45k training epochs. However, the quality of the ratio reconstruction is very stable during training. The behavior is similar to the one displayed in figure 2, discussed in the next section.

The central panel of figure 1 (labeled “SC w/ RW”) displays the quality of the log-ratio reconstruction by direct comparison with the true log-ratio $\tau_{\bar{c}} = \log r(x,\bar{c})$. The reconstruction is more accurate than the one, shown on the left panel with label “SC no RW”, of the neural network trained in ref. [5] with the same Monte Carlo statistics but using independent data sets for $c = 0$ and for $c = \bar{c}$. Using reweighted Monte Carlo data for training is evidently beneficial.

The visual comparison between the left and the middle panels of figure 1 conclusively shows that a more accurate reconstruction of the distribution ratio is possible using reweighted training data. On the other hand, the comparison does not offer a quantitative measure of the advantages of reweighting, ultimately because it is unclear, a priori, which accuracy is needed for a satisfactory reconstruction of the ratio. As explained in section 2.2.1, a useful quantitative performance metric can be defined only in relation to the actual experimental conditions in which the reconstructed ratio will be employed for statistical inference. We

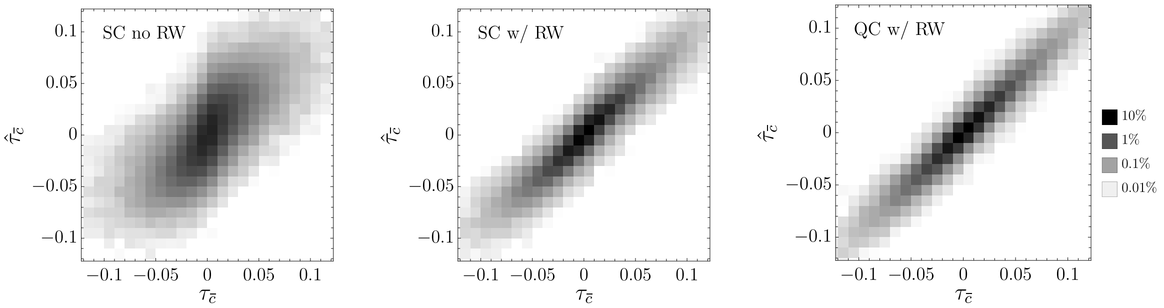


Figure 1. Density histograms displaying the correlation between the true $\tau_{\bar{c}}$ and the reconstructed $\hat{\tau}_{\bar{c}}$ obtained with three different strategy. From left to right: the simple classifier (SC) trained in ref. [5] on independent Monte Carlo samples for $c = \bar{c}$ and $c = 0$; the simple classifier trained with reweighted (RW) data; the quadratic classifier (QC) trained as explained in section 3.1.2. The $\bar{c}$ point has $G_\varphi = 0$ and $G_W = 10^{-2}\,\text{TeV}^{-2}$.

target the HL-LHC collider, with an integrated luminosity of 3 ab^{-1}. The ideal Monte Carlo predicts a total cross section $\sigma(0) = 1.06$ fb, and $\sigma(\bar{c}) = \sigma(0) + 4.0$ ab, corresponding to an expected data statistics $N(0) = 3180$, within the SM, and to $N(\bar{c}) - N(0) = 12$. Using this information we compute the median p-value $\hat{p}(\bar{c})$ defined in section 2.2.1. Ten million ideal Monte Carlo data are employed to determine $\hat{p}(\bar{c})$ using the binned strategy described in section 2.2.2. An error is assigned to $\hat{p}(\bar{c})$ by repeating the determination on ten partitions of the 10M data set and computing the standard deviation. The model from ref. [5], corresponding to the left panel of figure 1, has $\hat{p}(\bar{c}) = (3.8 \pm 0.3)\%$, while using reweighted events (middle panel) we reach $\hat{p}(\bar{c}) = (1.78 \pm 0.04)\%$.

The right panel of figure 1 (labeled "QC w/ RW") displays the reconstruction performances of the quadratic classifier strategy defined in sections 2.1.2 and 2.1.3, trained with reweighted data with the benchmark hyper-parameters described in the following section. The reconstruction further improves in comparison with the "SC w/ RW" setup, but the p-value only drops by 20%: $\hat{p}(\bar{c}) = (1.40 \pm 0.04)\%$. A 20% improvement of the p-value is appreciable but modest, especially in comparison with the reduction of a factor more than two of the "SC w/ RW" p-value relative to the p-value in the "SC no RW" setup. We discussed in section 2.2.1 that the saturation of the p-value is expected to occur when the reconstructed distribution ratio is so close to the true ratio that the test of hypothesis performed with the reconstructed likelihood attains nearly-optimal performances. The optimal median p-value, obtained using the knowledge of the true distribution ratio that is available for ideal data, is $p(\bar{c}) = (1.12 \pm 0.04)\%$. The "SC w/ RW" p-value is quite close to the optimal p-value. The improvement in the ratio reconstruction that is achieved in the "QC w/ RW" setup can lower the p-value, but obviously not push it below the optimal p-value. The performance gain is thus unavoidably moderate.

The slight gap in performances between the "SC w/ RW" and the "QC w/ RW" models probably emerges from the combination of two factors. First, the quadratic classifier strategy is advantageous because it exploits the quadratic dependence of the ratio on the Wilson coefficients in order to learn the ratio simultaneously from several different points in the c parameter space. Second, no hyper-parameters optimization was performed in the case of the simple classifier, while the quadratic classifier hyper-parameters are optimized as discussed in section 3.1.3. Regardless of performance, the simple classifier is not a viable strategy for the EFT likelihood learning. We will thus not investigate it further, nor optimize its performances, and devote the rest of this section to the quadratic classifier.

3.1.2 The quadratic classifier

Statistical inference requires the knowledge of the distribution ratio $r(x;c)$ as a function of the Wilson coefficients c. This can be learned efficiently by exploiting the known (quadratic) dependence of the ratio on c as explained in sections 2.1.2 and 2.1.3. The implementation of this strategy on ideal Monte Carlo data is presented below.

We are interested in the dependence of r on the two Wilson coefficients $c = (G_\varphi, G_W)$, however, we start from the simpler one-dimensional problem of learning the ratio in the direction of each of the two coefficients, setting the other one to zero. This is a valid starting point in general because the terms in the distribution ratio that are linear in the Wilson coefficients are typically harder to reconstruct, and often more important for the sensitivity because they account for the leading corrections to the SM distributions. It is thus convenient to study and optimize the reconstruction of each of them separately in the different one-dimensional problems, where they contribute individually. Furthermore, separate trainings in the direction of each Wilson coefficient are the first step of the efficient parallelizable learning strategy described in section 2.1.3. Learning the dependence of the ratio on G_φ turns out — as anticipated — to be rather simple in the case at hand. We thus describe the one-dimensional learning problem only in the $G_\varphi = 0$ direction and consider non-vanishing G_φ only for the two-dimensional study.

We employ the one-dimensional parametrization in eq. (2.12) using two feed-forward neural networks to model the two coefficient functions $\rho(x)$ and $\theta(x)$. The input is pre-processed by the transformation in eq. (3.2). In the benchmark configuration, $(10, 24, 24, 1)$ architecture and sigmoid activation functions are considered for the two networks. No activation function is applied to the output of the network. The parametrization is inserted in the classification function defined by eq. (2.6), out of which the loss function in eq. (2.7) is constructed. At the end of training, the reconstructed distribution ratio $\hat{r}(x;c)$ is given by

$$\hat{r}(x;c) = \mathcal{P}(\rho(x), \theta(x); c) = 1 + 2\,c\,\rho(x)\,\sin\theta(x) + c^2\rho^2(x)\,, \tag{3.5}$$

as a function of the Wilson coefficient $c = G_W$.

The quadratic classifier loss function (2.7) depends on a list $\mathcal{C} = \{\bar{c}^{(1)}, \ldots, \bar{c}^{(\kappa)}\}$ of points in the Wilson coefficient parameter space. These points can be freely chosen and they are among the hyper-parameters associated with the training of the quadratic classifier. In the benchmark configuration, we use the following values

$$\mathcal{C} = \{\bar{G}_W^{(1)}, \ldots, \bar{G}_W^{(6)}\} = \{\pm 5, \pm 2.5, \pm 1.25\} \times 10^{-2}\,\text{TeV}^{-2}\,. \tag{3.6}$$

The criteria for selecting these hyper-parameters, and the (very mild) effect of departures from the benchmark choice, are discussed in section 3.1.3.

Training is performed on 3M re-weighted Monte Carlo events, and 3M independent data are used for validation. We adopt the 2-step training strategy with decreasing learning rate

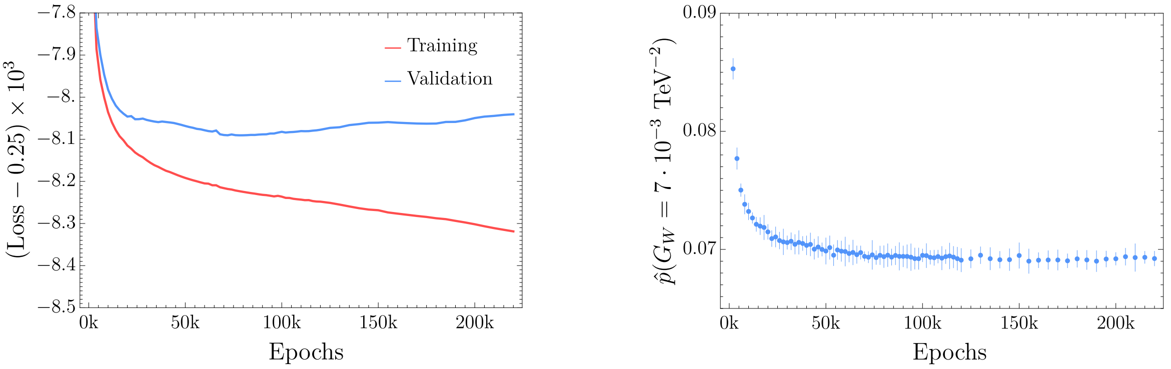


Figure 2. Evolution of the training loss and validation loss (left panel) and of the median p-value evaluated at $G_W = 7 \cdot 10^{-3}\,\mathrm{TeV}^{-2}$ (right panel) for the benchmark network.

described in section 3.1.1. The left panel of figure 2 shows the evolution of the loss function during training. Actually, the one reported in the figure is a re-scaled loss function, obtained by dividing eq. (2.7) by the sum of all the weights that appear in the equation, namely by $w_{\mathrm{e}}(\bar{c}) + w_{\mathrm{e}}(0)$ summed over e and over $\bar{c} \in \mathcal{C}$. With this normalization, an indecisive classification function $f = 1/2$ gives a loss of exactly $1/4$. Our classification function will be close to $1/2$ because the distribution ratio is close to one. Plotting the normalized loss minus 0.25, times a large factor, thus provides a better representation of the training evolution. Notice that the training and the validation loss are normalized separately. This avoids the emergence of spurious differences due to the different total weight of the training and of the validation data sets. Also notice that the normalized loss and not the original loss (2.7) is passed to the optimizer in order to prevent possible issues associated with a loss that is numerically very far from unity.

Training is run for many epochs, waiting for an increase in the validation loss that signals overfitting. The trained model configuration is the one that minimizes the validation loss. In figure 2, the minimum is attained after around 65k training epochs. We also evaluate, during training, the median p-value $\hat{p}(\bar{c})$ for a representative value $c = \bar{c}$ of the Wilson coefficient. This is chosen to be close to the conventional threshold for exclusion of 5%. $G_W = 7 \cdot 10^{-3}\,\mathrm{TeV}^{-2}$ is employed in figure 2. The p-value computed at run time is obtained with the skew-normal approximation of the test statistic distribution described in section 2.2.2. In figure 2 we report instead the binned determination of the p-value, evaluated off-line on the saved models. The two determinations are actually in good agreement in the specific case under examination.

Figure 2 displays a remarkable stability of the p-value during training, which extends deeply in the overfitted region where the validation loss (moderately) increases. We also observe a precise correspondence between the minimum of the validation loss — at 65k epochs — and the onset of the p-value stability region. The figure also shows that good performances could have been obtained also with less training epochs: after 10k epochs the p-value is only 10% larger than at the end of training. We are not limited by the training time because training for 1000 epochs takes around 1 minute in the benchmark setup, on the NVIDIA A30 GPU we used to produce our results. A reduction of the number of training epochs could however be considered in computationally more demanding problems.

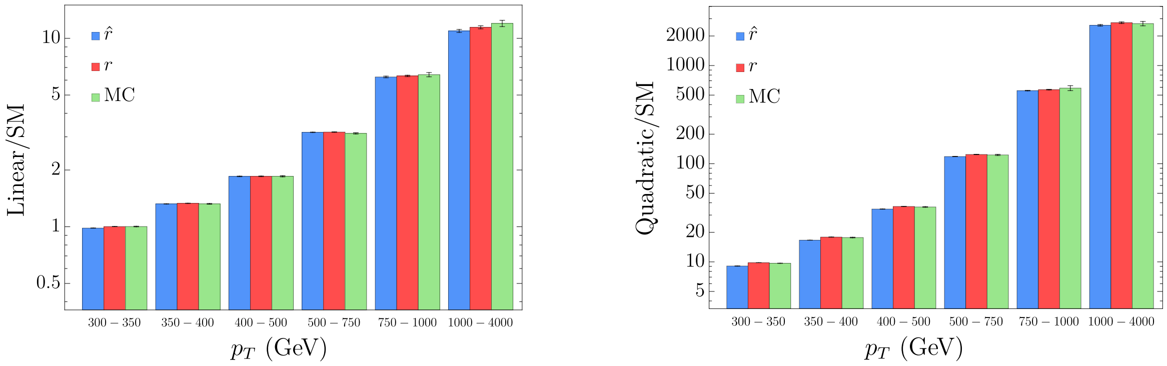


Figure 3. Linear (left) and quadratic (right) contributions of the $\mathcal{O}_W$ operator to the cross section in p_T bins, in the ideal setup. The direct determination from Monte Carlo data, by eq. (2.15) is shown in green. The determination from the reconstructed ratio (see (2.16)) in the benchmark configuration is shown in blue. The one from the exact ratio is in red. Angular cuts $|\cos\varphi_{W/Z}| > 0.6$ are applied.

We discussed in section 2.2 that the median p-value is our prime performance metric. It is particularly powerful in the case of ideal data because the optimal p-value can be computed owing to the knowledge of the r ratio. For the value $G_W = 7\cdot 10^{-3}\,\text{TeV}^{-2}$, considered in figure 2, the optimal p-value is $(6.5 \pm 0.1)\%$, while the quadratic classifier in the benchmark configuration gives $(6.95 \pm 0.04)\%$. We can thus conclude that the quadratic classifier in the benchmark configuration has reached effectively optimal performances and no further improvement is needed in the reconstruction of the distribution ratio. However, the r ratio and in turn the optimal p-value is never known in realistic problems. A more direct validation of the quality of the distribution ratio reconstruction must thus accompany the calculation of the median p-value.

The simplest test of the r-ratio reconstruction quality is to compare the predictions for the cross section in bins obtained from the reconstructed ratio $\hat{r}(x;c)$, as in eq. (2.16), with direct Monte Carlo estimates. Particularly significant binned distributions must be selected for the comparison. In figure 3 we consider the transverse momentum of the Z boson, p_T, with cuts $|\cos\varphi_{W/Z}| > 0.6$ on the W and Z decay angles. The selections are needed to access — in order to validate its reconstruction — the growing-with-energy linear term in G_W, which cancels in observable integrated over the angles. The linear and quadratic contributions to the distribution, relative to the SM, are reported separately in the figure. The reconstructed predictions (in blue) are in good agreement with the Monte Carlo predictions (in green). Both predictions are affected by errors due to the finite Monte Carlo statistics. Errors are obtained by splitting the test data in 10 subsets of 1M points each, repeating the determination of the observables on each subset and computing the standard deviation. The figures also reports the predictions obtained using, in eq. (2.16), the true r in place of the reconstructed $\hat{r}$. The perfect agreement with the Monte Carlo predictions provides a cross-check and supports the credibility of our error estimate.

We can also monitor the quality of the reconstruction by exploiting the peculiar property of $\mathcal{T}_c$, defined in eq. (2.20) as the logarithm of the cross section ratio in bins of $\hat{\tau}_c(x) = \log\hat{r}(x;c)$. We consider $c = G_W = 7\cdot 10^{-3}$ TeV^{-2} and we employ 50 bins, defined in such a way that each bin contains 1/50 of the SM cross section. A finer binning could provide a more accurate

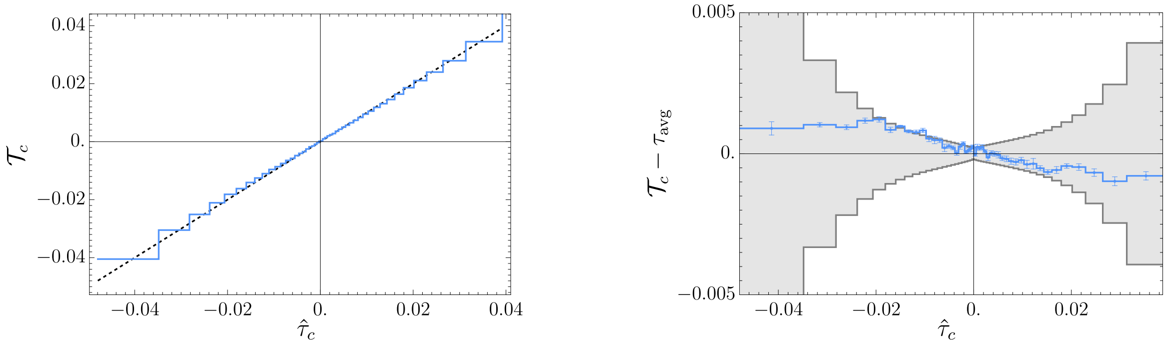
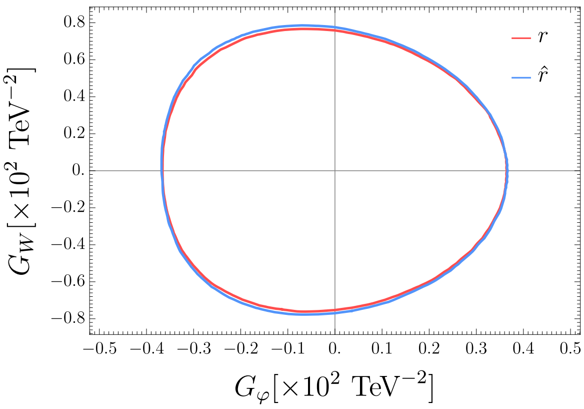


Figure 4. The $\mathcal{T}_c$ quantity of eq. (2.20), for the reconstructed $\hat{r}$ ratio obtained with the quadratic classifier in the benchmark configuration, with ideal data. $G_W = 7 \cdot 10^{-3}$ TeV^{-2} is considered. The right panel illustrates the approximate validity of the bounds in eq. (2.22), as explained in the main text.

test of the quality of the reconstruction, but it would compromise the accuracy of the $\mathcal{T}_c$ prediction due to the finite Monte Carlo statistics. The results, shown on the left panel of figure 4, display the approximate relation $\mathcal{T}_c = \tau_{\text{avg}}$ — with $\tau_{\text{avg}} = (\hat{\tau}_+ + \hat{\tau}_-)/2$ the center of the bin — that signals a good agreement of the reconstructed ratio with the true ratio. The right panel of the figure verifies the approximate validity of the bounds in eq. (2.22) by plotting $\mathcal{T}_c - \tau_{\text{avg}}$ overlaid with the intervals $[(\hat{\tau}_- - \hat{\tau}_+)/2, (\hat{\tau}_+ - \hat{\tau}_-)/2]$, for each bin, represented as a shadowed region. We see that $\mathcal{T}_c$ is often far from the center of the interval. It falls preferentially close to the upper or to the lower extreme. This does not signal a poor agreement between the reconstructed and the true ratio: we verified that the same behavior is observed using the true log-ratio $\tau_c(x) = \log r(x; c)$ instead of the reconstructed one for the calculation of $\mathcal{T}_c$. This is because the τ_c distribution is sharply peaked at zero. The cross section integral — see for instance eq. (2.21)—is thus dominated by the lower or upper extreme of the integration region for positive and negative τ, respectively. We thus preferentially saturate the lower/upper extreme of eq. (2.22) for positive/negative τ, precisely like we see happening for the reconstructed $\hat{\tau}_c$, in the figure. Because of these considerations, the only indication of a discrepancy between the reconstructed and the true ratio are those bins where the bound is strictly violated. There are very few such bins on the right panel of figure 4.

Since we got satisfactory performances on the one-dimensional problem, we can now address the complete task of learning the distribution ratio in the plane $c = (G_\varphi, G_W)$. We use the two-dimensional parametrization of eq. (2.10), employing 5 feed-forward neural networks to model the two ρ and the three θ coefficient functions. For the modeling of ρ and θ in the G_W direction we use the $(10, 24, 24, 1)$ architecture, which was found to perform well on the one-dimensional problem. The architecture for ρ and θ in the G_φ direction could in principle have been chosen differently, following a hyper-parameters optimization on the one-dimensional G_φ problem. However, since learning the dependence of the ratio on G_φ is a very simple task as previously discussed, no specific optimization is required and the $(10, 24, 24, 1)$ architecture is chosen for simplicity. The same architecture is also used for the third θ network, which describes the $G_\varphi G_W$ mixed term. The parametrization is inserted in the classification function (2.6) and eventually in the loss function (2.7). After training, the reconstructed $\hat{r}(x; c)$ is obtained from the parametrization similarly to eq. (3.5).



G_W	r	$[-0.752(2), 0.756(3)]$
	$\hat{r}$	$[-0.770(2), 0.773(2)]$
G_φ	r	$[-0.3657(4), 0.3642(3)]$
	$\hat{r}$	$[-0.3671(3), 0.3654(3)]$
$G_W = G_\varphi$	r	$[-0.3409(6), 0.3207(4)]$
	$\hat{r}$	$[-0.3436(6), 0.3228(4)]$
$G_W = -G_\varphi$	r	$[-0.3232(4), 0.3433(5)]$
	$\hat{r}$	$[-0.3257(4), 0.3477(5)]$

Figure 5. LEFT: contour lines $\hat{p}(c) = 5\%$ in the two-dimensional parameter space, for ideal data. The blue contours are obtained with the reconstruction of the r-ratio obtained by the quadratic classifier trained in two dimensions. The red contours are obtained using the exact ratio. RIGHT: reach at NLO on G_φ and G_W in 4 directions on the plane. The bounds are given in units of $10^{-2}\,\mathrm{TeV}^{-2}$.

The training points in the $\mathcal{C}$ set are formed by the G_W values in eq. (3.6) setting $G_\varphi = 0$, plus $G_W = 0$ points with G_φ in the set

$$G_\varphi \in \{\pm 2.5, \pm 1.25, \pm 0.625\} \times 10^{-2}\,\mathrm{TeV}^{-2}\,, \tag{3.7}$$

and six additional points with both G_W and G_φ non-vanishing along the diagonal of the grid formed by the two lists of values.

An alternative approach to the determination of the two-dimensional ratio (see section 2.1.3) is to learn the diagonal ρ and θ networks in one dimension, and to perform two-dimensional training only to train the mixed θ network. This computationally convenient strategy is not needed for our analysis, because our resources are sufficient to train the five neural networks at once directly in the two-dimensional setup. One might have expected a more accurate reconstruction of the ratio with two individual one-dimensional trainings, but in fact no such advantage has been observed in the case under examination. We have verified that one-dimensional trainings give essentially identical results as training directly in two dimension. In particular, the same p-values are found in the two cases along the single-coefficient lines $G_\varphi = 0$ or $G_W = 0$.

Several studies were performed to validate the accuracy of the distribution ratio reconstruction along different directions of the 2-dimensional parameter space. The performances are similar — or better, along the G_φ direction — than the ones previously described for G_W in the one-dimensional case. The corresponding plots are not reported for brevity. In essence, we find that the quadratic classifier attains a basically perfect reconstruction of the distribution ratio, that guarantees nearly-optimal statistical performances. This is shown in figure 5 by drawing 95% exclusion contours in the (G_φ, G_W) plane. The plot is obtained by computing and interpolating $\hat{p}(c)$ on a grid of points, and drawing the $\hat{p}(c) = 5\%$ contours. The red contour is obtained using the exact r ratio in place of the reconstructed ratio $\hat{r}$. By the Neyman-Pearson Lemma, the one that employs the exact ratio is the most powerful hypothesis test, namely the one with the smallest median expected p-value that

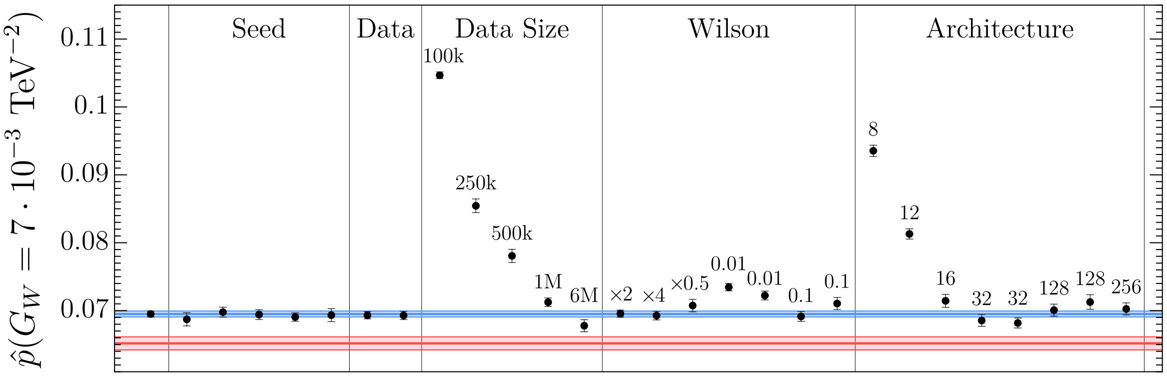


Figure 6. Median p-value for $G_W = 7 \cdot 10^{-3}$ TeV^{-2} in the configurations described in the main text. The blue band is the result obtained in the benchmark configuration. The red band is the optimal p-value. Training is repeated with two different seeds for the 32 and 128 neurons architectures.

in turn corresponds to the optimal (smallest) exclusion contour. The contour obtained with the reconstructed ratio, in blue, essentially coincides with the optimal contour. The 95 CL exclusion reach is also reported in the table in the right panel of figure 5 in different directions of the parameter space.

3.1.3 Hyper-parameters selection

The benchmark choice of the hyper-parameters described in the previous section results from an optimization of the performances. The median p-value was used as the prime indicator, but other performance metrics such as those in figures 3 and 4 were also employed, finding good correlation with the median p-value. The optimization was performed in the one-dimensional setup with $G_\varphi = 0$ as previously explained. In this section, we report some of the results of this hyper-parameters scan. In particular, we study the dependence of the performances on each individual hyper-parameter, keeping the other ones fixed to the benchmark values. The performances are measured by the median p-value at $G_W = 7 \cdot 10^{-3}$ TeV^{-2}. The results are reported in figure 6 and compared with the benchmark p-value, reported as a blue band.

We consider variations of the benchmark architecture, employing two hidden layers with a variable number of neurons. We observe, in figure 6, the expected saturation of the performances. Small networks are not expressive enough to accommodate an accurate modeling of the true distribution ratio. Large networks can instead model the distribution ratio accurately, fully exploiting the information that is present in the training data. On the other hand, large networks are more exposed to overfitting, which might prevent an accurate learning. Indeed, the training of large networks behaves differently from the one of the benchmark networks, displayed in figure 2: overfitting is more pronounced and it is associated with a degradation of the p-value. Using the model configuration that minimizes the validation loss eliminates this issue and delivers good performances also with rather large networks, as the figure shows.

In figure 6 we also study the dependence on the number of Monte Carlo data points used for training. A very significant degradation of the performances is observed only with a significant reduction of the training data set size relative to the benchmark of 3M points. The

p-value is only slightly larger than the benchmark p-value using 1M training points. Using 6M points the p-value is smaller than the benchmark p-value, but the gain in performance is too limited to justify doubling the training sample size. The benchmark choice of 3M points is almost equally effective and in fact, very good results would have been obtained already with 1M points.

We also report in figure 6 two technical checks of stability. The "Seed" column displays the p-value obtained by repeating training in the benchmark configuration using 5 different random seeds for the initialization of the neural network parameters. In the "Data" column we repeat training on two independent data sets. Since they are independent of the seed and of the specific data set used for training, our results are robust and reproducible.

In the "Wilson" column we study the dependence on the values of the Wilson coefficients used for training, namely on the set of values, $\mathcal{C}$, that defines the loss function of the quadratic classifier in eq. (2.1). The values used in the benchmark configuration — reported in eq. (3.6)— were selected with a very simple criterion. At 95% CL, the analysis is sensitive to values of G_W that are of the order of 10^{-2} TeV^{-2}. We should thus prioritize an accurate reconstruction of the ratio when G_W in this range, because much larger values will be in any case easy to exclude while much lower values are invisible. We thus use for training values of G_W that are of this order of magnitude, with a substantial spacing among them in order to provide more information about the dependence on the ratio on G_W. In order to study the sensitivity to the specific benchmark choice, the values (3.6) used in the benchmark configuration were raised by a factor of 2 or of 4, or divided by 2. The results — labeled as "×2", "×4" and "×0.5" in figure 6, are essentially identical to the one in the benchmark configuration. Employing values of G_W that are loosely close to the reach thus ensures good performances and no precise optimization is needed. Another advantage of considering relatively small values of the Wilson coefficients for the training is the fact that the corresponding reweighting factors are typically of order one, thus minimizing possible accuracy issues with the weighted data-sets (see discussion in the Introduction). We also stress that, although weighted events are essential for an efficient learning algorithm, unweighted sets could be used for the final statistical analysis, thus solving possible issues due to a lower statistical accuracy.

We also tried a simpler training scheme that employs only 2 distinct equal and opposite values of G_W instead of 6 values as in the benchmark configuration. The results for $G_W = \pm 0.1$ TeV^{-2} and $G_W = \pm 0.01$ TeV^{-2} are reported in the figure. Two results are reported for each value by employing different random seeds. A very mild degradation of the performances is observed, showing that employing more than 2 values of G_W for training as in the benchmark configuration is beneficial, but only marginally. We also tried to use $G_W = \pm 0.001$ TeV^{-2} and $G_W = \pm 1$ TeV^{-2}, much below and much above the reach, respectively. In the first case, training was found to proceed too slowly, which was expected because the gradients of the loss function become small if the training points are too close to the SM $G_W = 0$ point. In the second case, strong overfitting was observed, probably due to the fact that if G_W is overly large the loss function is dominated by the high-energy tail of the training data points, where the EFT contribution from the quadratic terms dominates over the SM. In summary, a good stability of the performances is observed under variations of the values of the Wilson coefficients used for training, provided they are of the order of the expected reach and not vastly above or below.

These results should be contrasted with the findings of ref. [5], where the quadratic classifier was trained on independent Monte Carlo samples generated at each point in $\mathcal{C}$ without using reweighting. In that setup, the training points are a major factor controlling the performances. They must be chosen by balancing two competing criteria. First, the values of the Wilson coefficients used for training must be large enough to modify the SM distribution considerably and typically well above the sensitivity reach. Otherwise, their effects would be too small to be seen by comparing independent Monte Carlo data sets. We explained in the Introduction and in section 2.1.1—and we verified in section 3.1.1—that small departures from the SM are on the contrary effectively reconstructed if reweighted data are used for training. Therefore, we do not need to employ large values of the Wilson coefficients for training. The second criterion for the selection of the Wilson coefficients in the setup of ref. [5] is to avoid overly large values. Otherwise, the contribution to the cross section would be dominated by the quadratic term. The effect of the linear term would be a small correction and could not be learned. Reweighted training is sensitive to small effects, therefore also this second criterion is less relevant to our methodology. The optimal Wilson coefficient that obeys the above-mentioned criteria depends on the phase space region because the size of the EFT contribution to the cross section scales, in particular, with the energy of the process. The optimal Wilson coefficients in three p_T regions were identified in ref. [5] and used for training producing good performances. However, the need of optimizing the choice of the training points is an obstruction to the systematic deployment of the methodology of ref. [5]. This optimization is not required because the performances are weakly sensitive to the training points with the methodology of the present paper.

3.2 Results at NLO

We finally describe our results on NLO simulations of the ZW process. Reweighted Monte Carlo samples are generated using the MadGraph software suite, as detailed in section A.2. Like in the ideal case, 3M points are used for training and 3M independent points are used to compute the validation loss. Only 3M points rather than 10M are employed for testing. Ten independent observable variables x characterize the NLO events, listed in eq. (A.21). These are pre-processed with the transformation

$$\begin{aligned} x \;\to\; \Big\{ & \log[s/\mathrm{GeV}^2],\, \Theta,\, \theta_Z,\, \theta_W,\, \log[p_T/\mathrm{GeV}], \\ & \log[1+p_{T,\mathrm{ZW}}/\mathrm{GeV}],\, Q,\, \ell_Z,\, \ell_W\ \sin\varphi_Z,\, \sin\varphi_W,\, \cos\varphi_Z,\, \cos\varphi_W \Big\}, \end{aligned} \tag{3.8}$$

and normalized to zero mean and unit variance on the training data set by a normalization layer in the neural networks. The discrete labels ℓ_Z and ℓ_W that describe the flavor of the leptons from the Z and the W boson decays assume the numerical values of 1 or 0 for muons and electrons, respectively.

The implementation of the quadratic classifier method proceeds as explained in section 3.1.2 and no significant difference is observed — neither in the behavior of the training process nor in the performances — associated with the usage of NLO rather than ideal Monte Carlo data. This is interesting because NLO Monte Carlo data sets contain events with negative weights. These events give a negative contribution to the loss function that might

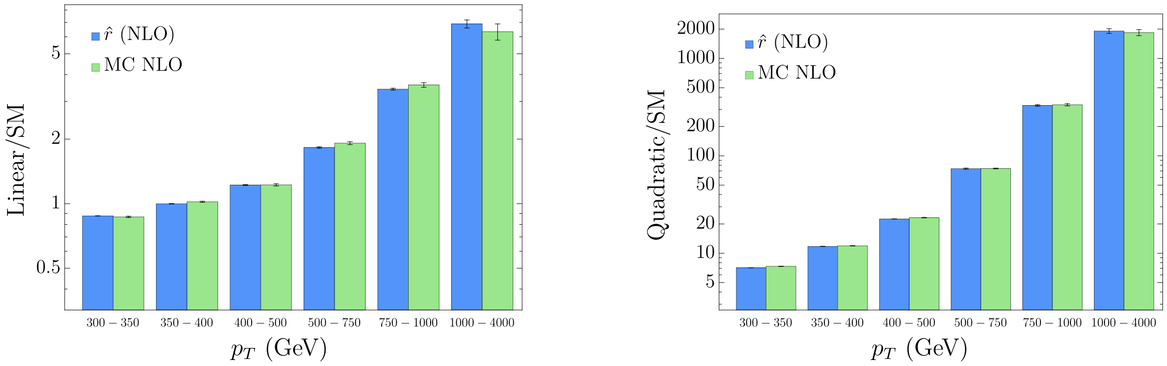


Figure 7. Linear (left) and quadratic (right) contributions of the $\mathcal{O}_W$ operator to the cross section in p_T bins, using NLO simulations. The direct determination from Monte Carlo data, by eq. (2.15) is shown in green. The determination from the reconstructed ratio (see (2.16)) is shown in blue. Cuts $|\cos\varphi_{W/Z}| > 0.6$ and $p_{T,\mathrm{ZW}}/p_T < 0.5$ are applied.

encourage overfitting, raising a concern about the possibility of applying our methodology to realistic NLO simulation. Our methodology is instead found to behave on NLO data as well as on ideal data, where no negative weights are present. It should be noted that currently available NLO generators like MadGraph produce a small fraction of negative-weights events: they are 3% of the total in the Monte Carlo data sets under consideration. A larger fraction of negative weights that might be obtained with a non-optimized generator, or in a different process, could in principle invalidate this conclusion.

Since no strong difference is found in comparison with the ideal results presented in the previous section, we illustrate our findings directly on the full two-dimensional setup where the distribution ratio — parametrized as in eq. (2.11)—is learned with a single training as a function of the two Wilson coefficients G_φ and G_W. All the hyper-parameters are set to the benchmark values we used for ideal data in the previous section, apart from the architecture of the neural networks that is now $(13, 32, 32, 1)$ to account for the larger dimensionality of the NLO input features. The performances of the trained model are shown in figures 7 and 8.

Figure 7 displays the ability of the trained model to reproduce the linear and quadratic contributions of the $\mathcal{O}_W$ operator to the cross section in p_T bins and with cuts $|\cos\varphi_{W/Z}| > 0.6$ on the azimuthal decay angles and $p_{T,\mathrm{ZW}}/p_T < 0.5$ on the total transverse momentum. These cuts emphasize the leading growing-with-energy contribution of the $\mathcal{O}_W$ operator. The same plot is shown in figure 3 for ideal data and the agreement between the reconstructed (blue) and the Monte Carlo (green) predictions is at the same level as in figure 3. Figure 3 also shows the results obtained using the knowledge of the true distribution ratio r, which is available for ideal simulations and not available for NLO simulations. Figure 8 displays the approximate linear behavior of the $\mathcal{T}_c$ variable and the validity of the bounds in eq. (2.22), at the point c in the parameter space where $G_W = 7 \cdot 10^{-3}$ TeV^{-2} and $G_\varphi = 0$. The plot is obtained in the same way as figure 4 for ideal data. The results confirm that a good reconstruction of the distribution ratio is obtained also at NLO.

Finally, the left panel of figure 9 displays the 5% contours of the median p-value $\hat{p}(c)$. For its determination, we proceeded as explained in section 2.2.2 and in the previous section. The right panel of the figure summarized the reach along four directions in the parameter space.

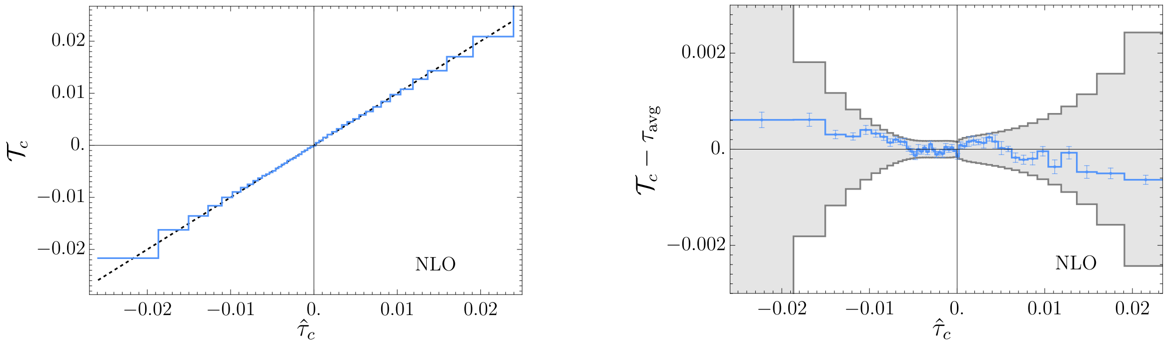
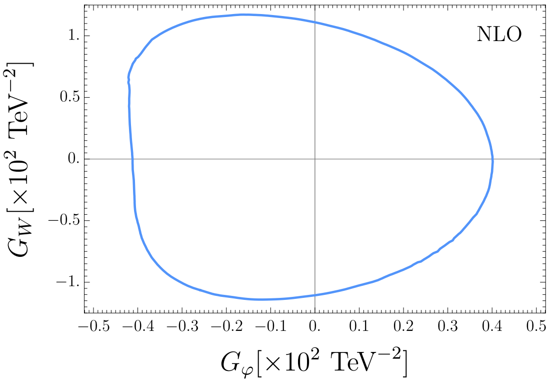


Figure 8. The $\mathcal{T}_c$ quantity of eq. (2.20), for the reconstructed $\hat{r}$ ratio obtained with the quadratic classifier, with NLO data. The point $G_W = 7 \cdot 10^{-3}$ TeV^{-2} and $G_\varphi = 0$ is considered. The right panel illustrates the approximate validity of the bounds in eq. (2.22), as explained in the main text.



G_W	$[-1.106(6), 1.108(7)]$
G_φ	$[-0.412(1), 0.401(1)]$
$G_W = G_\varphi$	$[-0.405(1), 0.366(1)]$
$G_W = -G_\varphi$	$[-0.376(1), 0.419(1)]$

Figure 9. LEFT: contour lines $\hat{p}(c) = 5\%$ in the two-dimensional parameter space, for NLO data. RIGHT: reach at NLO on G_φ and G_W in the 4 directions of the 2D plane. The bounds are given in units of 10^{-2} TeV^{-2}.

4 A proposal for exclusions

We explained in the Introduction and in section 2.2.1 that our methodology is not tied to the specific task of the statistical analysis one eventually aims at performing on the experimental data, nor to the specific (frequentist or Bayesian) methodology that will be adopted for the analysis. Our ultimate goal is to extract a good approximation of the distribution ratio (1.3), to be later used for any analysis aimed at approaching optimality exploiting the accurate knowledge of the ratio and in turn of the likelihood. On the other hand, it is worth discussing and addressing the challenges in the design and in the practical implementation of one such nearly-optimal analysis. We focus in particular on the design of a classical test of hypothesis aimed at excluding the existence of EFT operators with a given pre-specified value $c \neq 0$ for their Wilson coefficients (the H_0 hypothesis) in favor of the SM value $c = 0$ (the H_1 hypothesis). A Bayesian approach to the same problem is described in section 4.4.

As explained in section 2.2.1, the Neyman-Pearson Lemma guarantees the optimality of the test based on the likelihood ratio test statistic $t_c(\mathcal{D})$ defined in eq. (2.26). Since we are able to reconstruct the likelihood log-ratio, a natural option would be to employ the reconstructed test statistic $\hat{t}_c(\mathcal{D})$ defined in eq. (2.27) for the actual statistical analysis on the observed data. The test statistic depends on the variable $\hat{r}_c(x)$, which is the logarithm of the reconstructed distribution ratio (2.17). If the reconstruction is accurate, we are guaranteed that this test attains optimal performances, as we also verified in section 3. This theoretical guarantee of optimality is precisely the reason why the median p-value of the test based on $\hat{t}_c(\mathcal{D})$ is used for a robust assessment of the quality of the distribution ratio reconstruction. On the other hand, it would be difficult to apply this test to real data. In this section, we define an alternative test that could be more viable in practice and with expected performances that are not far from optimality.

The main factor that limits the applicability of the $\hat{t}_c(\mathcal{D})$-based test — or of the test based on the exact $t_c(\mathcal{D})$, if available — is that the distribution of the test statistic variable is in general not known. On the other hand, we need the distribution in the null $c \neq 0$ hypothesis in order to obtain the p-value. We also need the distribution in the alternative $c = 0$ hypothesis, if we want to quantify the expected sensitivity of the test. The determination of the distributions has to be performed numerically, and this requires generating many toy instances of the $\mathcal{D}$ data set, which is numerically expensive. In section 2.2.2 we addressed this issue by discretizing the $\hat{r}_c(x)$ variable in a set of bins. This enabled us — in eq. (2.30)—to express $\hat{t}_c(\mathcal{D})$ in terms of the counts $\mathcal{N}_{\rm b}$ in each bin, which are Poisson-distributed and efficient to sample for the generation of toys. Inspired by that result, we consider the possibility of defining an analysis strategy based on binning the $\hat{r}_c(x)$ variable.

4.1 The optimal binned test

On general grounds, a *binned* analysis is one that divides the x space in a set of non-overlapping bins $\rm b \in B$ and computes the number of points in $\mathcal{D}$, $\mathcal{N}_{\rm b}$, that fall in each bin. The counts, $\mathcal{D}_{\rm B} = \{\mathcal{N}_{\rm b}, \forall\, {\rm b} \in {\rm B}\}$, are the aggregate data used for statistical inference, in place of the original data set $\mathcal{D}$. Importantly enough, the $\mathcal{N}_{\rm b}$ variables are independent Poisson distributions with an expected, $N_{\rm b}(c) = L\,\Delta\sigma_{\rm b}$, that can be accurately calculated with the Monte Carlo, at least if the number of bins is not extremely large. Therefore, we have complete analytical access to the distribution of the binned data. On the contrary, an analysis based on variables such as the reconstructed (2.27) or the exact (2.26) log-ratio, which depend directly on the observables vector $x \in \mathcal{D}$, is called *unbinned*. The fact that we do not have access to the x data distribution makes unbinned analyses generically more difficult. Binned analyses are easier and way more common in LHC data analysis practice.

The analytical knowledge of the distribution of the binned data $\mathcal{D}_{\rm B}$ enables the deployment of the optimal Neyman-Pearson strategy for hypothesis testing. This test would be based on the negative of the log-ratio between the binned likelihoods in the $c \neq 0$ and in the $c = 0$ hypotheses. Specifically, the optimal test statistic for binned data would be

$$-2\,\log\frac{\mathfrak{L}(c;\mathcal{D}_{\rm B})}{\mathfrak{L}(0;\mathcal{D}_{\rm B})} = 2\sum_{\rm b\in B}\left[N_{\rm b}(c) - N_{\rm b}(0) - \mathcal{N}_{\rm b}\log\frac{N_{\rm b}(c)}{N_{\rm b}(0)}\right]\,. \tag{4.1}$$

One should be careful here with the notion of optimality. The guarantee of optimality of the Neyman-Pearson Lemma is obviously tied to the nature of the data upon which the statistical inference is based. The complete data set $\mathcal{D}$ contains all the information that is collected by the experiment, therefore the (unbinned) likelihood ratio associated with the $\mathcal{D}$ data is the truly optimal test statistic. The binned likelihood ratio in eq. (4.1) would be the optimal variable if the counts $\mathcal{D}_{\rm B}$ were the only available experimental information, but its performances are in general sub-optimal because binning reduces information.

Consider however defining the bins in terms of the variable $\tau_c(x)$, which is the logarithm of the true distribution ratio. Namely, the bins are defined by the conditions $\tau_c(x) \in (\tau_{\rm b-1}, \tau_{\rm b})$, with $\rm b = 1, \ldots, n_{\rm bin}$, $\tau_0 = -\infty$ and $\tau_{n_{\rm bin}} = +\infty$ in order to cover the whole x space. In this case, the binned likelihood log-ratio (4.1) is easily shown to coincide, when the bins are infinitely narrow, with the unbinned likelihood ratio $t_c(\mathcal{D})$ defined in eq. (2.26). In fact, if the bins are very narrow, the $\tau_c(x)$ variable is well-approximated over the entire x space by the following piecewise constant function

$$\tau_c(x) \;\simeq\; \left\{\log\frac{\Delta\sigma_{\rm b}(c)}{\Delta\sigma_{\rm b}(0)}\,,\ \text{for } x\ \text{ s.t. } \tau_c(x)\in(\tau_{\rm b-1},\,\tau_{\rm b})\right\}\,, \tag{4.2}$$

where $\Delta\sigma_{\rm b}(c)$ is the cross section in the bin. This approximation holds because $\tau_c(x)$ is the logarithm of the distribution ratio and thus it can be computed as the logarithm of the ratio of the cross sections integrated in a narrow bin.[6] Notice that the value of τ_c in the bin could have also been approximated by the center of the bin, which in fact coincides with the log cross section ratio up to finite bin effects as we proved in eq. (2.21). Inserting eq. (4.2) into the definition (2.26) of $t_c(\mathcal{D})$, we find

$$t_c(\mathcal{D}) \simeq 2\left[N(c) - N(0) - \sum_{\rm b\in B}\mathcal{N}_{\rm b}\log\frac{N_{\rm b}(c)}{N_{\rm b}(0)}\right]. \tag{4.3}$$

This expression coincides with eq. (4.1) since $\sum_{\rm b} N_{\rm b} = N$. Therefore, the binned likelihood ratio (4.1) approaches the optimal test statistics, and hence it is nearly-optimal, if the binning is performed on the $\tau_c(x)$ variable and if a sufficiently narrow binning is employed. Clearly, we do not have access to the exact $\tau_c(x)$ variable. The reconstructed $\hat{\tau}_c(x)$ variable will be used instead to define the bins. Optimality will thus be attained only if the reconstruction is accurate.

The development of methodologies aimed, like the one of the presented paper, at reconstructing the distribution log-ratio $\tau_c(x)$ is often motivated by the need of accessing the unbinned log-likelihood ratio for optimal statistical inference. The previous results however show that binned data contain, up to finite-binning effects, the exact same amount of information as the original data set $\mathcal{D}$. In particular, we have seen that the optimal log-likelihood ratio test on binned data has the same sensitivity as the optimal unbinned test. An unbinned approach is thus not essential to attain near-optimality. The reconstruction of $\tau_c(x)$ is instead essential because it is crucial that the binning is performed on the $\tau_c(x)$ variable or a good approximation thereof. Binning any other variable would lead to sub-optimal performances even for infinitely narrow binning, barring the possibility of binning all the individual components of the data vector x, which is not feasible in more than two or

[6] The first and the last bins extend up to infinity and hence are not narrow. Eq. (4.2) still holds if τ_c is bounded. If it is not, extreme tails of the τ_c distribution can be excluded from the analysis of the data without sensitivity loss because they are not populated by the finite data statistics.

three dimensions. The $\tau_c(x)$ variable needs to be reconstructed because it brings the whole information about the parameters of interest that is available in the data. It can be used for an unbinned analysis, or for a binned analysis like we propose.

4.2 Maximum likelihood and Asymptotic formulas

The implementation of the optimal binned hypothesis test defined by eq. (4.1) still requires drawing toys to determine the test statistic distributions empirically. Poisson-distributed toys are numerically cheap to generate at one fixed point c in the parameter space, but repeating the toys generation and the determination of the distribution point-by-point in the parameter space can easily become unfeasible if the number of parameters is large. The relevant parameters here are not only the parameters of interest c. Imperfections in the Monte Carlo predictions due to the approximate knowledge of the underlying physical model, or of the modeling of the detector response, need to be incorporated in the analysis in the form of nuisance parameters, denoted as ν. Determining the dependence of the likelihood on the nuisance parameters is rather standard practice in LHC data analysis. It requires estimating the dependence of the number of expected events in each bin, $N_{\rm b}(c,\nu)$, not only on the parameters of interest but also on the nuisance parameters. This determination requires dedicated Monte Carlo events, and it can be demanding to achieve but is feasible. It is instead not feasible to scan this large parameter space for the empirical determination of the test statistic distribution, which in general depends on the parameters as the test statistic variable does. Practical LHC data analysis thus needs to rely on analytical approximations of the distributions. These approximations are typically valid in the large-sample — or, Asymptotic — limit. Therefore, they are called Asymptotic formulas.

No Asymptotic formula exists to model the distribution of the likelihood log-ratio test statistic (4.1). We thus need to employ a different variable, which in turn defines a different (and sub-optimal) hypothesis test. Following the standard LHC data analysis practice [40], we use the *maximum likelihood ratio* test statistics, which is defined as follows. Consider an open curve that interpolates from the SM point $c = 0$ to the point $c \neq 0$ that we are interested in testing, in the space of the parameters of interest. For definiteness, we choose a straight line but other options would be also viable as depicted in figure 10. Be γ_μ an explicit parametrization of the curve, where $\mu \in \mathbb{R}$ would correspond to the "signal strength" parameter in the notation of ref. [40]. The curve γ_μ defines a one-parameter family of statistical hypotheses for the distribution of the data, within the larger space span by the Wilson coefficients c, which interpolates between H_0 and H_1. The maximum likelihood test statistics is -2 times the logarithm of the ratio between the likelihood at c and the maximum of the likelihood along the γ_μ curve, namely

$$t_c^{\rm ML}(\mathcal{D}_{\rm B}) = -2\,\log\frac{\mathfrak{L}(c;\mathcal{D}_{\rm B})}{\mathfrak{L}(\gamma_{\widehat{\mu}};\mathcal{D}_{\rm B})} = 2\sum_{\rm b\in B}\left[N_{\rm b}(c) - N_{\rm b}(\gamma_{\widehat{\mu}}) - \mathcal{N}_{\rm b}\log\frac{N_{\rm b}(c)}{N_{\rm b}(\gamma_{\widehat{\mu}})}\right]\,. \tag{4.4}$$

The expression is similar to the likelihood log-ratio test statistics (4.1). The difference is that the likelihood in the denominator is not computed at the SM nor at any other pre-specified point of the parameter space. It is computed at the point $c = \gamma_{\widehat{\mu}}$ that maximizes the likelihood along the curve for the specific instance, $\mathcal{D}_{\rm B}$, of the binned data set under

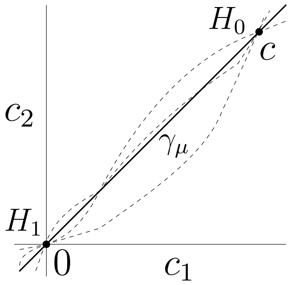


Figure 10. Schematic representation of the interpolation between the H_0 hypothesis (the point c in the parameter space) and the H_1 hypothesis (the origin, $c = 0$).

examination. The maximal point on the curve, located at $\mu = \widehat{\mu}$, is thus an implicit function of the data: $\widehat{\mu} = \widehat{\mu}(\mathcal{D}_{\rm B})$.

Unlike the likelihood log-ratio (4.1), the maximum likelihood test statistics (4.4) is endowed with Asymptotic formulas. Under the null hypothesis $c \neq 0$, it approaches a χ^2 with one degree of freedom in the Asymptotic limit. Under the alternative hypothesis $c = 0$ it approaches a non-central χ^2 with one degree of freedom. These formulas allow us to implement the test of hypothesis based on the maximum likelihood ratio test statistic without having to resort to toy data. The p-value associated to a given observed value t of the $t_c^{\rm ML}(\mathcal{D}_{\rm B})$ test statistic, defined as in eq. (2.24), is simply given by

$$p_c(t) = p[t] = 1 - {\rm CDF}_{\chi^2_1}(t)\,, \tag{4.5}$$

where CDF denotes the cumulative distribution function. Notice that the association between the value of the test statistic and the p-value, namely the $p_c(t)$ function, depends on c in general because the distribution of the test statistics does depend on c. The function is instead independent of c in this particular case because the Asymptotic formula for the distribution is a χ^2 independently of c.

We can also use approximations to compute the expected p-value under the SM hypothesis $c = 0$. Several methods were considered in ref. [40]. The one that is most commonly employed is based on the so-called "Asimov" data set, $\mathcal{D}_{\rm B}^{\mathcal{A}}$. The Asimov data set is one where all the observables are exactly equal to their expected value. In the case at hand

$$\mathcal{D}_{\rm B}^{\mathcal{A}} = \{N_{\rm b}(0), \forall\, {\rm b} \in {\rm B}\}\,, \tag{4.6}$$

where $N_{\rm b}(0)$ is the expected number of events in each bin as predicted by the SM $c = 0$ hypothesis. Ref. [40] proposed to estimate the median of the $t_c^{\rm ML}$ variable as the value it assumes on the Asimov data set. Consequently, the median p-value can be estimated as

$$p_{\rm med}^{\rm ML}(c) = p\left[t_c^{\rm ML}(\mathcal{D}_{\rm B}^{\mathcal{A}})\right]\,. \tag{4.7}$$

A few comments are in order. First, the validity of the Asymptotic formulas relies on the availability of a large data statistics. Giving for granted that the total expected number of events is large, for the validity of the Asymptotic formulas we still have to ensure that the number of expected events in each bin is also large. Experience reveals that 10 events are sufficient for the Asymptotic formulas to hold accurately.[7] This limits the maximum number of bins that can be employed in the analysis, entailing some loss of sensitivity because, as we have seen, the data binned in the $\hat{\tau}_c$ variable contain the same amount of information as the original data only in the limit of infinitely narrow binning. However, in practice, we do not expect this to be the main source of degradation in comparison with the optimal test sensitivity. In fact, it should be stressed that the one based on maximum likelihood is a different test of hypothesis than the one based on the likelihood ratio test statistic (4.1), and hence it is intrinsically sub-optimal. Extensive experience in statistical practice ensures that its performances are typically not far from optimal, but as far as we know this fact is not guaranteed by any rigorous result. Whenever possible, it is useful to compare its sensitivity with the one of the optimal test for a few representative points of the parameter space in order to get an idea of the degradation. Finally we also mention that the use of the maximum likelihood test is essential in the presence of nuisance parameters, whose impact can not be taken into account through a likelihood ratio statistical test à la Neyman-Pearson.

The role played by the curve γ_μ along which the maximum of the likelihood is computed also deserves to be described in detail. First, we notice that the shape of the curve plays no role in sensitivity projections based on the Asymptotic formulas and on the Asimov trick. Namely, the median expected p-value in eq. (4.7) is independent of the shape of the curve. The $p(t)$ function is universally provided by the cumulative of the χ^2, and obviously independent of the γ_μ curve. The value of the test statistic on the Asimov data set is also independent of the curve, because of the following. On the Asimov data set, the logarithm of the likelihood along the γ_μ curve is

$$\log \mathfrak{L}(\gamma_\mu; \mathcal{D}_{\rm B}^{\mathcal{A}}) = \sum_{\rm b \in B} \left[-N_{\rm b}(\gamma_\mu) + N_{\rm b}(0) \log N_{\rm b}(\gamma_\mu)\right] + \text{ const.}\,, \tag{4.8}$$

and attains its absolute maximum, in each bin, when $N_{\rm b}(\gamma_\mu) = N_{\rm b}(c)$. The point on the curve that maximizes the likelihood is thus the SM point, $\gamma_{\widehat{\mu}} = 0$. Evaluating the maximum likelihood test statistic (4.4) on the Asimov data set thus gives

$$t_c^{\rm ML}(\mathcal{D}_{\rm B}^{\mathcal{A}}) = -2 \,\log \frac{\mathfrak{L}(c; \mathcal{D}_{\rm B}^{\mathcal{A}})}{\mathfrak{L}(0; \mathcal{D}_{\rm B}^{\mathcal{A}})} = 2 \sum_{\rm b \in B} \left[N_{\rm b}(c) - N_{\rm b}(0) - N_{\rm b}(0) \log \frac{N_{\rm b}(c)}{N_{\rm b}(0)} \right]\,, \tag{4.9}$$

regardless of the shape of the γ_μ curve.

The shape of the curve does affect, on the contrary, the determination of $t_c^{\rm ML}$ on the real data, and in turn it affects the p-value that will be obtained in the analysis of the data collected by the experiment. Each curve formally defines a different test of hypothesis. Since all these tests have the same expected sensitivity as previously discussed, they are all equally valid and each of them could be applied to the data. The most sensible line of action would be to select, prior to the experiment, a simple curve like a straight line and stick to that choice. Otherwise, one could combine the p-values obtained with different curves easily, because all curves have the same expected sensitivity.

One might wonder why considering a curve, rather than interpolating between c and the origin with a higher-dimensional surface. A seemingly natural choice would be to employ

[7] The validity of the Asymptotic formulas can be directly related, in the case at hand, to the accuracy of the Gaussian approximation for the Poisson distribution, which is good for more than 5 or 10 expected events.

in fact the whole c space for the interpolation. In the present paper we are dealing with two Wilson coefficients: one might have considered defining the maximum likelihood ratio test statistic (4.4) by minimizing over the whole Wilson coefficient plane rather than on the curve γ_μ. Asymptotic formulas are available for the test statistic distribution even if the maximization is performed on a multidimensional family of hypotheses. They involve χ^2 distributions with a number of degrees of freedom that is equal to the dimensionality of the family. In particular, a central χ^2 with two degrees of freedom would have been obtained for the distribution of the test statistic in the null $c \neq 0$ hypothesis, if the maximization was performed on the entire plane. In eq. (4.5) we would thus have encountered the cumulative of the χ^2 with two rather than one degree of freedom. On the contrary, the determination of the median expected value of the test statistics, estimated by the Asimov trick, would have remained the same as in eq. (4.9). The χ^2 with two degrees of freedom is broader than the χ^2 with one degree of freedom. Therefore, the median expected p-value in eq. (4.7) would be higher if the maximization was performed on the two variables rather than on the one-dimensional curve. We thus discard this possibility because it would produce a test that is less performant and farther from optimality.

Finally, we notice that the inclusion of nuisance parameters in the maximum likelihood ratio framework is conceptually straightforward, following ref. [40]. It merely amounts to maximizing over the nuisance parameters, independently, the likelihoods in the numerator and in the denominator of eq. (4.4). The Asymptotic χ^2 formulas still hold. On the Asimov data set — defined at the central value of the nuisance parameters $\nu = 0$, namely $\mathcal{N}_{\rm b} = N_{\rm b}(0,0)$ — the likelihood is still maximal at the SM point $c = \gamma_{\widehat{\mu}} = 0$ and with central-value nuisance. The maximum likelihood-ratio test statistics on the Asimov data is similar to eq. (4.9), though it involves a potentially costly minimization over the nuisance parameters. Namely, in the presence of nuisance parameters ν, eq. (4.9) becomes

$$t_c^{\rm ML}(\mathcal{D}_{\rm B}^{\mathcal{A}}) = 2\min_{\nu}\left\{\sum_{\rm b\in B}\left[N_{\rm b}(c,\nu) - N_{\rm b}(0,0) - N_{\rm b}(0,0)\log\frac{N_{\rm b}(c,\nu)}{N_{\rm b}(0,0)} - \log\frac{\mathfrak{L}(\nu)}{\mathfrak{L}(0)}\right]\right\}. \tag{4.10}$$

In the equation, $\mathfrak{L}(\nu)$ is the likelihood for the nuisance parameters as determined from measurements that are independent of the data under examination. This likelihood is often simply approximated with a multivariate Gaussian centered on the central-value nuisance configuration $\nu = 0$. The minimization to be performed in eq. (4.10) and other elements that are peculiar to our proposal could complicate its practical implementation in the presence of many nuisance parameters. This is discussed in the following section.

4.3 Implementation

The implementation of our proposal requires, in the first place, to select a suitable set of points $\tau_{\rm b}$ on the real axis for binning the $\hat{\tau}_c(x)$ variables. For a given chosen number of bins, $n_{\rm bin}$, these points are selected in such a way that all resulting bins have an equal cross section in the SM hypothesis, namely $\Delta\sigma_{\rm b}(0) = \sigma(0)/n_{\rm bin}$. The points are determined by evaluating $\hat{\tau}_c(x)$ on the Monte Carlo data set, sorting the list and scanning through it cumulating the SM weights $w_{\rm e}(0)$ until when integer multiples of the target cross section $\sigma(0)/n_{\rm bin}$ are reached. We use $n_{\rm bin} = 300$ because this produces — both with the ideal and with the NLO

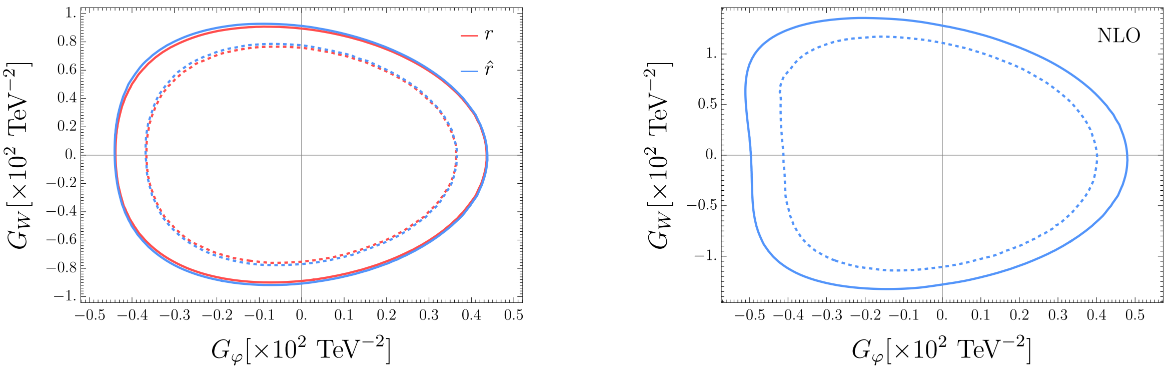


Figure 11. Contours of $p^{\rm ML}_{\rm med}(c) = 5\%$ (i.e., $t^{\rm ML}_c(\mathcal{D}^{\mathcal{A}}_{\rm B}) = 3.84$) in the Wilson coefficient plane $c = (G_\varphi, G_W)$ for ideal (left panel) and NLO (right panel) simulations. The contours obtained using the exact τ_c rather than the reconstructed $\hat{\tau}_c$ are displayed in red color on the left panel. The dashed lines are the contours from figures 5 and 9.

predictions — around 10 expected events in each bin, ensuring the validity of the Asymptotic formulas. Once the binning points $\tau_{\rm b}$ are determined, the cross sections and the expected number of events for non-vanishing c are determined by summing the $w_{\rm e}(c)$ weights of the events that fall in each bin. Finally, the Asimov estimator, $t^{\rm ML}_c(\mathcal{D}^{\mathcal{A}}_{\rm B})$, is computed by eq. (4.9). The median p-value $p^{\rm ML}_{\rm med}$ is given by eq. (4.7).

We applied this algorithm to draw 95% CL expected exclusion contours in the plane of the two Wilson coefficients $c = (G_\varphi, G_W)$ that we studied in section 3. The results are displayed as continuous blue lines in figure 11. The left/right panel of the figure shows the results for ideal/NLO simulations. The reconstructed distribution ratio is obtained from the parameterized classifier trained as explained in sections 3.1 and 3.2 for ideal and NLO simulations, respectively, using benchmark hyper-parameters. The plots are obtained computing $t^{\rm ML}_c(\mathcal{D}^{\mathcal{A}}_{\rm B})$ on a grid of points on the plane. The 95% CL contour is the region where $p^{\rm ML}_{\rm med}(c) = 5\%$, i.e. where $t^{\rm ML}_c(\mathcal{D}^{\mathcal{A}}_{\rm B}) = 3.84$ because of the χ^2 formula. The contours are drawn from an interpolation of the grid. The exact distribution ratio is available for ideal data. We can thus run the previous algorithm but use the exact $\tau_c(x)$ in place of the reconstructed $\hat{\tau}_c(x)$. The result is shown with a red continuous line on the left panel of the figure. The excellent agreement with the blue contour provides an additional cross-check of the accuracy of the distribution ratio reconstruction, though arguably a less powerful one than those we performed in section 3 because it is based on an inherently less sensitive hypothesis test methodology than the likelihood-ratio test we employ as a performance metric.

The figure also displays, in dashed, the contours previously reported in figures 5 and 9. These are the 5% contours of the optimal Neyman-Pearson test p-value. They provide tighter exclusion bounds than the test based on the maximum likelihood ratio, as expected. However, the contours are relatively close, showing that the maximum likelihood ratio test is not vastly sub-optimal, in the case at hand.

As a cross-check, we verify in figure 12 the validity of the Asymptotic and Asimov approximations of the maximum likelihood ratio test sensitivity. For this study, we generated Poisson-distributed toy data sets for the binned data under the null and under the alternative

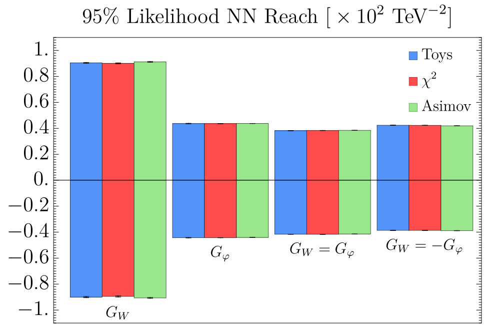


Figure 12. Upper and lower exclusion reach at 95% CL along four directions in the Wilson coefficient plane $c = (G_\varphi, G_W)$. Blue bars are obtained empirically with toy data sets. Red bands make use of the Asymptotic χ^2 approximation for the test statistics distribution under the null hypothesis, while the median p-value is determined empirically on the toys. The green band employs both the Asymptotic and the Asimov approximations as in eq. (4.7).

hypotheses, and we computed $t_c^{\mathrm{ML}}(\mathcal{D}_{\mathrm{B}})$ from the definition (4.4), performing the maximization along the γ_μ curve. The curve is taken to be a straight line interpolating from $c = 0$ to the test point $c \neq 0$. We determine by interpolation the eight points on the c plane where the empirically-determined p-value equals 5%, along four directions in the plane. Ideal simulations are used for this study. The results are shown as blue bars in figure 12. The green bars are obtained using the Asymptotic χ^2 formula for the distribution under the null hypothesis and the Asimov data set for the determination of the median. They correspond to the intersection between the contour on the left panel of figure 11 and the four selected directions in the plane. The red bars in the figures employ instead the Asymptotic χ^2 formula, but estimate the median by toy experiments performed under the alternative hypothesis. The three sets of results are in excellent agreement, confirming the accuracy of the Asymptotic and Asimov approximations. The result was widely expected because these approximations are well-established and routinely employed for binned data with sufficient statistics in each bin.

Our proposal for setting exclusion limits is based on concepts and approximations that are deeply routed in the LHC statistical practice. However, it should be stressed that an analysis based on our proposal is rather different from a regular LHC binned exclusion analysis, and potentially more demanding computationally. In a regular binned analysis, both the variable or variables used for binning and the bins are pre-specified. Here instead they depend on the point c in the space of the parameters of interest that we are interested in probing. The $c = 0$ and $c \neq 0$ cross sections in each bin are thus computed once and for all in a regular analysis, while in our case they are computed point-by-point as previously explained, because we have access to the $\hat{\tau}_c(x)$ variable used for binning only after the relevant value of c is specified. This is the price to pay for attaining nearly-optimal sensitivity: the variable to be binned is optimized point-by-point in the c space.

The additional computational cost, in comparison with the cost of a regular binned analysis on real LHC data, is not easy to quantify. A few considerations are reported below. A realistic LHC analysis is affected by systematic uncertainties incorporated in a typically large number of nuisance parameters. As previously explained, these parameters need to be incorporated in the likelihood by determining their effect on the expected $N_{\rm b}(c,\nu)$ in each bin using dedicated Monte Carlo data sets. This calculation is not a significant addition to the total computational cost in a regular binned analysis, because it is performed once and for all in the pre-defined bins. The most costly part of the analysis is the minimization over the nuisance parameters, for each c, that has to be performed for the determination of the Asimov estimate of the median $t_c^{\rm ML}$ as in eq. (4.10). For our strategy, one needs to run the minimization, but also to first determine $N_{\rm b}(c,\nu)$ at each point in the c space. The relative computational cost of these two operations defines the additional cost of our proposal in comparison with a regular binned analysis. The Bayesian counterpart of minimization, i.e. the integration over the nuisance, would pose a similar challenge in the alternative approach described in the next section.

4.4 Bayes factor

The problem of comparing the agreement with data of two hypotheses H_0 and H_1 — in our case, the hypotheses that c assumes one given non-vanishing value, and the one that $c=0$, respectively — could be also addressed in a Bayesian framework without performing a test of hypothesis. In this case, one would need to compute the Bayes factor, which is the ratio of the posterior probabilities for the two hypotheses if the priors $P(H_1)$ and $P(H_0)$ are equal.[8] In the absence of nuisance parameters

$$\mathcal{B}_{10}(\mathcal{D}) = \frac{P(H_1|\mathcal{D})}{P(H_0|\mathcal{D})} = \frac{\mathfrak{L}(0;\mathcal{D})}{\mathfrak{L}(c;\mathcal{D})} = e^{-\lambda(c;\mathcal{D})} = e^{t_c(\mathcal{D})/2}\,, \tag{4.11}$$

with λ the unbinned log-likelihood ratio in eq. (1.2) and $t_c(\mathcal{D})$ the optimal Neyman-Pearson test statistic in eq. (2.26). A large value of $\mathcal{B}_{10}$, which corresponds to large and positive $t_c(\mathcal{D})$, signals poor agreement of the H_0 hypothesis with the data and in turn a low posterior H_0 probability.

If the exact distribution log-ratio $\tau_c(x)$ was known, we could simply evaluate $t_c(\mathcal{D})$, compute the Bayes factor (4.11) and, based on its value, draw conclusions on the plausibility of the null hypothesis H_0. This could be done on a representative data set drawn under the alternative $c=0$ hypotheses — or the median could be computed over $c=0$ toys — in order to determine the expected sensitivity of the analysis. However, we do not have access to $\tau_c(x)$, but only to the reconstructed $\hat{\tau}_c(x)$ and in turn to the reconstructed likelihood log-ratio $\hat{t}_c(\mathcal{D})$ in eq. (2.27). We can monitor the agreement between $\hat{\tau}_c$ and τ_c, but we have no way to control quantitatively the departure of the reconstructed Byes factor $\hat{\mathcal{B}}_{10} = \exp(\hat{t}_c/2)$ from the true Bayes factor.

It should be noted that the remark above is an obstruction to employ the reconstructed Bayes factor for the analysis of the actual data, but an even more severe obstruction to employ it as a metric to assess the performances of the distribution log-ratio reconstruction. Errors on the determination of the log ratio can make $\hat{\mathcal{B}}_{10}$ larger or smaller than the true $\mathcal{B}_{10}$. In particular, any overfitting that makes $\hat{\tau}_c$ large and negative produces a large and positive $\hat{t}_c$ and in turn large $\hat{\mathcal{B}}_{10}$. Therefore, monitoring $\hat{\mathcal{B}}_{10}$ would not give any information on the quality of the reconstruction and a model with larger $\hat{\mathcal{B}}_{10}$ could easily be worse than

[8] In general, $P(H_1|\mathcal{D})/P(H_0|\mathcal{D}) = \mathcal{B}_{10}P(H_1)/P(H_0)$.

one with smaller $\hat{\mathcal{B}}_{10}$. The classical Neyman-Pearson median p-value is instead sensitive to the quality of the reconstruction owing to the Neyman-Pearson Lemma, which guarantees it to attain its absolute minimum when the true ratio is reconstructed, and obliges it to be larger if the reconstruction is less accurate. This is the reason why a classical hypothesis test was pursued in section 2.2.1 to define a performance metric and Bayesian inference methods were not considered for that task.

We could, on the other hand, employ Bayesian inference on the analysis of the actual data, by proceeding as follows. The distribution of the binned data is known exactly, hence the Bayes factor can be exactly computed and reads

$$\log \mathcal{B}_{10}^{\rm B}(\mathcal{D}_{\rm B}) = \sum_{\rm b \in B} \left[N_{\rm b}(c) - N_{\rm b}(0) - \mathcal{N}_{\rm b} \log \frac{N_{\rm b}(c)}{N_{\rm b}(0)} \right] . \tag{4.12}$$

We have seen that the binned data contain the same information as the original data set $\mathcal{D}$ if the binning is performed on the $\hat{\tau}_c(x)$ variable, provided this variable is a good approximation of $\tau_c(x)$. Errors in the $\tau_c(x)$ reconstruction can undermine the optimality of the analysis, entailing some loss of information to occur in the process of binning. However, they do not undermine eq. (4.12) as a consistent calculation of the Bayes factor for the binned data, because the binned data are truly Poisson-distributed regardless of the variable that is employed for binning.

It is interesting to notice that this binned Bayesian approach to exclusions is in fact very similar to its frequentist counterpart based on the maximum likelihood ratio that we defined in the previous section. Consider estimating the value of the Bayes factor that is expected to be observed if the data are distributed as predicted by the SM, i.e. for $c = 0$. A reasonable estimate is to evaluate the Bayes factor on the Asimov data set defined by eq. (4.6). This gives

$$\log \mathcal{B}_{01}^{\rm B}(\mathcal{D}_{\rm B}^{\mathcal{A}}) = \sum_{\rm b \in B} \left[N_{\rm b}(c) - N_{\rm b}(0) - N_{\rm b}(0) \log \frac{N_{\rm b}(c)}{N_{\rm b}(0)} \right] . \tag{4.13}$$

This expression is equal to half the maximum likelihood ratio test statistics evaluated on the Asimov set, as computed in eq. (4.9). The $t_c^{\rm ML}(\mathcal{D}_{\rm B}^{\mathcal{A}}) = 3.84$ contour lines in figure 11, which correspond to 95% CL frequentist exclusions, are thus also contours where the expected Bayes factor equals $\exp(3.84/2) = 6.8$. The two approaches thus give the same results up to the conventional choice of the threshold for exclusion. On the other hand, notice that there is no exact correspondence between the value of $t_c^{\rm ML}$ that will be observed on the truly collected data set. Furthermore, nuisance parameters are treated differently in the Bayesian and in the frequentist frameworks leading to potentially different findings in the two cases.

5 Conclusions and outlook

We studied the advantages of employing reweighted Monte Carlo events for learning the dependence of the data distribution on the parameters of interest. We focused in particular on the Wilson coefficients of EFT interaction operators, in the perspective of their extensive investigation with the LHC and the forthcoming HL-LHC collider data. In section 2.1 we defined our methodology. This extends our previous proposal in ref. [5] to exploit reweighted

Monte Carlo data sets for training. In section 3 we studied the performances of the new methodology on the same benchmark problems of ref. [5], enabling direct comparisons.

Our first finding is that reweighted training data enable an accurate learning of the distribution ratio even when the effect of the Wilson coefficients on the distribution is a small correction to the SM prediction. This can be crucial in order to learn effects — within or outside the EFT context — that cannot be parametrized and require point-by-point learning with realistic values of the parameters, for which the effects are indeed small. Furthermore, even if the effects can be parametrized like in the quadratic classifier setup, the capability of reweighted training to capture small effects lies at the heart of the other major advantage with respect to the methodology of ref. [5]: the weak sensitivity of the performances to the choice of the values of the Wilson coefficients used for the training of the model (see section 3.1.3). The method of [5] is on the contrary very sensitive to this choice. Even if the origin of this sensitivity is understood, and criteria are defined in ref. [5] for an optimal choice, the need of optimizing these hyper-parameters makes the proposal of ref. [5] harder to implement systematically or to automate. For the methodology of the present paper instead, a loose criterion of selecting the training points close to the sensitivity reach of the experiment is enough for accurate learning.

In the table on the right panel of figure 5 we report (in the "$\hat{r}$" entries) 95% CL expected single-operator limits for the Wilson coefficient G_W and G_φ on ideal simulations. They are essentially identical in all directions of the plane to the limits obtained using the knowledge of the exact r ratio (the "r" entries of the table). On the contrary, in ref. [5], a slight gap in performances with the optimal reach could be observed in the G_W direction.[9] While the sensitivity improvement is marginal, it should be noted that a total of 12M training points were used in ref. [5], while only 3M are employed here. Furthermore, we saw in section 3.1.3 that almost identical performances would have been obtained using only 1M training points. This shows that reweighted training enables a more accurate learning with fewer points.

All the advantages of reweighting listed above are specific to the problem of learning the distribution ratio. On the other hand, our methodology also benefits from the generic advantage of reweighting, which is to enable predictions in the entire space of the parameters of interest using one single Monte Carlo sample generated at one point.

In the paper we also advanced, in section 2.2, the design of performance metrics to assess the quality of the distribution ratio reconstruction. These metrics are extensions of ideas from ref. [5], which however could be efficiently evaluated and employed systematically only thanks to the fast and accurate Monte Carlo predictions obtained with reweighting. The main performance indicator is the median p-value for the exclusion of EFT interactions. Its main advantage is to assess the quality of the distribution ratio reconstruction in relation with the actual experimental setup where it will be eventually employed for statistical inference. The saturation of this quantity towards the absolute minimum — which can be only reached with perfect reconstruction — signals that the quality of the reconstruction is sufficient for optimal statistical inference using the data of the experiment under consideration. This

[9] The relevant results are in table 1 (specifically, the "QC" entries) of ref. [5]. Notice that the optimal reach was estimated in ref. [5] using the near-Gaussian approximation rather than using bins (see section 2.2.2). This results in a slight discrepancy between the "ME" limits of ref. [5] and the "r" limits of figure 5.

defines a stopping criterion for the quality of the reconstruction, which should otherwise be indefinitely improved with larger neural networks and training data sets until when computationally feasible.

Finally, in section 4 we described how the learned distribution ratio could be used in practice for statistical inference on real data, or for sensitivity projections. We focused in particular on the task of setting exclusion limits point-by-point in the Wilson coefficient parameter space by comparing the agreement with data of the SM hypothesis — where the EFT Wilson coefficients vanish — with the hypothesis that they assume a given non-vanishing value. The exclusion will be set where the SM agreement strongly exceeds the EFT agreement. We saw that the problem can be addressed in a robust and computationally manageable manner, without major sensitivity loss, by using the reconstructed ratio to bin the data. A Bayesian approach to the problem is found to give the same result — in terms of sensitivity projections, and in the absence of nuisance parameters — of a classical approach that exploits Asymptotic formulas and the Asimov data set.

It should be noted that our ratio could be also used to reconstruct the unbinned likelihood ratio and employed for a global EFT fit, like in ref. [10]. On the other hand, performing a global fit is not necessarily the only or the most useful statistical inference task. The result of the fit depends on the prior expectations on the Wilson coefficients, which in turn depend on the microscopic physical model that gives origin to the effective interactions. Each microscopic physics scenario gives origin to one subset of all possible dimension-six EFT operators, or more generally it produces tight correlations between the Wilson coefficients of different operators. The correct prior that corresponds to each microscopic physics scenario should be used for a meaningful fit. A flat prior on all dimension-six operators, which is typically employed in EFT global fits, is not a valid prior because it does not correspond to the expectations of any known scenario: it never happens in concrete models that all operators are generated with comparable coefficients and no correlation among them. Point-by-point information in the Wilson coefficient parameter space, like the one we obtain with the strategy of section 4, is instead meaningful regardless of the microscopic origin of the EFT interaction operators. The information can be transferred easily to the parameter space of any specific microscopic physics model.

Our methodology could be used for two distinct classes of applications. One is performing systematic or even automated EFT sensitivity projections. The other, is performing real analyses of the LHC data. The perspectives for progress in these two directions are discussed in turn.

Automated nearly-optimal sensitivity projections would be extremely beneficial because the number of EFT interaction operators, as well as the number of promising LHC processes to detect their presence, are both large. Furthermore, each process is characterized by a typically large number of potential observables that could help probe some of the possible EFT operators. One out of many examples of non-trivial observables are the decay angles of the bosons in the ZW process we studied in the present paper, and in ref. [5]. Their measurement offers access to the leading contribution from the $\mathcal{O}_W$ operator, which would on the contrary cancel exactly if only the kinematical distributions of the two bosons were measured. We would have discovered this fact empirically, if we had not known it theoretically, by observing strong sensitivity improvement due to the inclusion of the decay angles in the automated nearly-optimal analysis, and/or by observing improvement in comparison with the sensitivity of a regular analysis binned, for instance, on the transverse momentum of the bosons. A similarly intricate interplay between observables and operators is definitely at work in many other cases that have not been subject to a careful theoretical investigation. The discovery of these phenomena enabled by automated sensitivity projections might eventually lead to the design of suitable observables and binning strategies by which nearly optimal sensitivity could be attained without using the learned distribution ratio in the actual analysis of the data. Notice however that this has not yet been achieved for the G_W operator in the ZW process: the binned analysis performed in ref. [5] including binning over the decay angles in order to access the relevant information about the decay of the bosons has a reach on G_W that is a factor of two larger than the nearly-optimal reach obtained with our methodology.

We consider that all elements are in place for the deployment of our methodology for automated sensitivity projections. The scale of the problems to be faced is not different from the one we addressed in the present paper with relatively limited computational resources. It is likely that one would like to extend the analysis to a larger number of interaction operators than the two operators we considered in the present paper. However, as discussed in section 2.1.3, the fact that the dependence of the distribution ratio on the Wilson coefficient is a quadratic polynomial implies that one can learn the dependence on any number of Wilson coefficients without the need of training with more than two non-vanishing coefficients. A parallelizable learning strategy was defined in section 2.1.3, based on which we expect to be able to deal with tens of Wilson coefficients with reasonably available computational resources. It is possible that further methodological advances will be achieved in the future. A direct performance comparison with other approaches [6, 10] would facilitate the emergence of new ideas for better and/or more efficient learning. However, the current state-of-the-art already enables the systematic exploration of several LHC processes and EFT operators.

The actual perspectives for systematically employing the learned distribution ratio in the analysis of the LHC data are instead still to be assessed. A major concern is the availability of synthetic data for training, validation and testing. For the present paper, we used a factor of few tens of thousands more data than the expected number of data points at the HL-LHC. The majority of them were employed for testing purposes, namely for the evaluation of performance metrics to assess the quality of the reconstruction ratio. Less accurate quality studies than the one we performed here would require fewer testing points, and we saw that also the size of the training data set could have been reduced without major sensitivity degradation. Still, it is expected that order thousands of times more statistics than the actual data would be required for the implementation of our strategy. Monte Carlo simulations that simulate the response of the detector accurately are computationally costly, potentially preventing the generation of such large data sets. Notice however that detector effects are often modeled accurately by simplified simulation tools like Delphes [52], which are fast to run. Training, validation and testing could be performed on Delphes samples, with the caveat that the learned ratio will be an approximation of the exact ratio that corresponds to the Delphes predictions, which is close but different from the true distribution ratio. Employing the Delphes ratio for statistical inference on the actual data will not entail a strong departure from optimality, if the Delphes simulation is accurate. Clearly it will be essential to employ a statistical inference methodology that is correct and internally consistent regardless of the agreement between the reconstructed and the true ratio, like the one we outlined in section 4. It is also essential that full detector simulations and not Delphes simulations are used to perform the analysis of the actual data and for sensitivity projections. The amount of simulated data required for this task is however expectedly not dissimilar from the one needed for a regular analysis. A realistic treatment of nuisance parameters will be also needed and this could challenge, for instance, the practical implementation of our proposal in section 4 for real data analysis. The viability of this strategy should be assessed for individual processes before drawing conclusions on the perspectives for the applicability of our strategy to real data. It should also be noted that the SM background data sets in real analyses of the LHC data often contain components that are estimated by data in a control region—extrapolated to the signal region by transfer functions—and not by running a Monte Carlo code. These data can be used like Monte Carlo data, that is included on top of the Monte Carlo simulations of the EFT effects, but they must be available with sufficient statistics and correctly model the distribution of all the relevant observables.

It is worth mentioning that strategies like ours aimed at learning the dependence of the data distribution on parameters can find applications also outside the context of EFT searches. One such application is to learn the dependence on nuisance parameters locally in the space of observables x. Our parametrized classifier approach is perfectly suited for this task because the nuisance parameters are precisely a parametrization of the effects of imperfections in the theoretical predictions. Their effects are relatively small, hence they are typically parametrized with a linear polynomial or other functional forms like an exponential scaling factor to enforce cross section positivity strictly. All these options can be readily implemented in a parametrized classifier as done in ref. [53] to learn the unbinned SM likelihood in order to incorporate the effects of nuisance in an unbinned strategy for agnostic new physics searches. When available, reweighted Monte Carlo data sets can facilitate the learning task. A clear case is the one of uncertainties on the parton distribution functions, which are incorporated in Monte Carlo event generators by event reweighting. It is expected that our methodology could learn the effect of the many nuisance parameters associated with parton distribution functions extremely efficiently.

Acknowledgments

A.W. is supported by the grant PID2020-115845GB- I00/AEI/10.13039/501100011033. G.P. was supported in part by the MIUR under contract 2017FMJFMW (PRIN2017).

A Fully-leptonic ZW at high energy

We consider the production of a Z and of a W boson at the 14 TeV LHC, with a lower cut of 300 GeV on their transverse momentum. The process is dominantly mediated by the reactions

$$u_L\,\overline{d}_L \to Z\,W^+ \;\text{ and }\; d_L\,\overline{u}_L \to Z\,W^- \,, \tag{A.1}$$

depending on the charge of the produced W boson. Leptonic decays are considered for both bosons, namely

$$Z \to \ell^+_\chi \ell^-_\chi \,, \qquad W^\pm \to \ell^\pm_L \overset{(-)}{\nu}_L \,, \tag{A.2}$$

where ℓ is either an electron or a muon and $\chi = L, R$ denotes fermion chirality.

The observable final state contains three leptons. The lepton pair that emerges from the decay of the Z boson can be identified as having opposite charge, same flavor (electron or muon) and, in case of ambiguity, the closest invariant mass to the one of the Z particle. The lepton chirality cannot be measured, hence the final states that are relevant for our analysis are labeled by a total of three discrete indices: the total charge $Q = \pm 1$ of the three leptons, which corresponds to the W boson charge; the flavor $\ell_Z = e, \mu$ of the leptons emerging from the decay of the Z; the flavor $\ell_W = e, \mu$ of the W boson decay. These three labels can be given as input to the neural network.

The neutrino escapes detection, however its momentum can be reconstructed starting from the total missing transverse energy of the event by imposing that it originates, together with an observable lepton, from the decay of a W boson. The reconstruction is performed by imposing that the invariant mass of the pair formed by the neutrino and the visible lepton is equal to the mass of the W particle. Since this defines a quadratic equation with two solutions, the neutrino momentum reconstruction is possible only up to a twofold ambiguity. We resolve this ambiguity by a random choice of the two solutions with equal probability on an event-by-event basis.[10] After the reconstruction, the following 7 variables are known (see [5] and references therein) to be useful to characterize the kinematics of the event

$$\{s,\, p_{T,\mathrm{ZW}},\, \Theta,\, \theta_W,\, \varphi_W,\, \theta_Z,\, \varphi_Z\}\,. \tag{A.3}$$

The first three variables describe the kinematics of the ZW pair, namely its total invariant mass squared, its total transverse momentum, and the angle of the Z boson in the ZW rest frame, relative to the direction of its boost in the lab frame. The remaining four variables are the polar (θ) and azimuthal (φ) decay angles of the bosons in their rest frames. The precise definition of these variables is detailed in [5, 46].

We aim, as in [5], at studying the two specific dimension-six interaction operators[11]

$$\mathcal{O}_\varphi = G_\varphi \left(\overline{Q}_L \sigma^a \gamma^\mu Q_L\right) (i H^\dagger \overleftrightarrow{D}_\mu H)\,, \qquad \mathcal{O}_W = G_W \varepsilon_{abc} W^{a\,\nu}_\mu W^{b\,\rho}_\nu W^{c\,\mu}_\rho \,, \tag{A.4}$$

by modeling their effects on the distributions with two different Monte Carlo event generators: the ideal and the NLO generators. The ideal generator is based on a simplified description of the process, relying on approximations that are not sufficiently accurate for the description of the actual experimental data. However, it produces distributions that are similar to those obtained with realistic simulations, and defines a learning problem with the same scale of complexity. At the same time, its simplifying assumptions enable the exact analytical

[10] The on-shell equation has no solution when the lepton transverse mass exceeds m_W because of experimental errors or because the W resonance was far from its mass-shell. In this case, the neutrino momentum is reconstructed by taking its rapidity to be equal to the one of the lepton. See [46] for details.

[11] We use the definition $H^\dagger \overleftrightarrow{D}_\mu H = H^\dagger D_\mu H - (D_\mu H)^\dagger H$.

calculation of the distribution ratio $r(x;c)$, to be compared with the reconstructed ratio $\hat{r}(x;c)$. This comparison provides an important validation of our methodology. The NLO generator offers instead an accurate description of the process, including radiative corrections from QCD at the one-loop order. This additional theoretical complexity does not change the complexity of the learning problem: the NLO distributions are similar to the ideal ones, the number of variables is (almost, see below) the same and no new relevant sub-process kicks in at NLO in the phase-space region that is relevant for the analysis. On the other hand, introducing NLO effects confronts our methodology with the novel challenge of negative weights. In fact, NLO corrections can be included in Monte Carlo data sets only by allowing for some of the events to have negative weight $w_{\rm e}(c) < 0$. Such events give a negative contribution to the loss function, encouraging overfitting. It is thus important to validate our methodology also on NLO data. No performance degradation will be observed, showing that our approach is perfectly suited to deal with negative-weight Monte Carlo data, at least if the fraction of negative weight is as small as for the state-of-the-art NLO generator that we employ in our study.

The next two sections provide a technical description of the ideal and of the NLO generators.

A.1 Ideal generator

The ideal generator relies, in the first place, on the narrow-width approximation for the decay of the two bosons. This enables us to factorize the scattering amplitude for the partonic processes (A.1) in terms of a "hard" amplitude describing the production of on-shell bosons, times the amplitudes that describe the decay of the bosons. The contributions of intermediate on-shell bosons with different helicities need to be summed up at the amplitude level, leading to the following expression for the differential cross section [5, 46]

$$d\sigma(\xi;c) = \sum_{h,h'} d\rho^{\rm hard}_{h_Z h_W h'_W h'_Z}\, d\rho^{Z}_{h_Z h'_Z}\, d\rho^{W}_{h_W h'_W}\,. \tag{A.5}$$

In the equation, $d\rho^{\rm hard}$ is the density matrix for polarized WZ production, namely

$$d\rho^{\rm hard}_{h_Z h_W h'_Z h'_W} = \frac{1}{24\,s}\mathcal{M}_{h_Z h_W}(\mathcal{M}_{h'_Z h'_W})^*\, d\Phi_{\rm ZW}\,, \tag{A.6}$$

where $d\Phi_{\rm ZW}$ is the phase-space factor. Since we are studying high-energy production, we can consider the limit of massless vector bosons. The non-vanishing production amplitudes in this limit read

$$\begin{aligned} \mathcal{M}_{00} &= -\frac{g^2\sin\bar{\Theta}}{2\sqrt{2}} - \sqrt{2}G_\varphi\bar{s}\sin\bar{\Theta}\,, &\qquad \mathcal{M}_{++} = \mathcal{M}_{--} &= \frac{3gc_{\rm w}G_W\bar{s}\sin\bar{\Theta}}{\sqrt{2}}\,, \\ \mathcal{M}_{-+} &= -\frac{g^2(s_{\rm w}^2 - 3\,c_{\rm w}^2\cos\bar{\Theta})}{3\sqrt{2}c_{\rm w}}\cot\frac{\bar{\Theta}}{2}\,, &\qquad \mathcal{M}_{+-} &= \frac{g^2(s_{\rm w}^2 - 3c_{\rm w}^2\cos\bar{\Theta})}{3\sqrt{2}c_{\rm w}}\tan\frac{\bar{\Theta}}{2}\,, \end{aligned} \tag{A.7}$$

where g is the $\mathrm{SU}(2)_L$ coupling, $c_{\rm w}$ and $s_{\rm w}$ are the cosine and the sine of the Weak angle.[12] We denote as $\bar{s}$ and $\bar{\Theta}$ the center of mass energy squared of the ZW system and the polar

[12] An overall factor equal to the cosine of the Cabibbo angle has not been reported for shortness. The ones above are the amplitudes for W^+ production. Those of the $d\bar{u} \to ZW^-$ process can be obtained with the formal substitutions $\bar{\Theta} \to -\bar{\Theta}$ and $s_{\rm w}^2 \to -s_{\rm w}^2$.

angle of the Z boson in order to distinguish them from the observable variables s and Θ. The relation between these quantities will be discussed later.

The density matrices $d\rho^{W,Z}$ describe the decay of the bosons. They take the generic form

$$d\rho^{V}_{h_V h'_V} = \frac{1}{2m_V\Gamma_V}\mathcal{A}_{h_V}\mathcal{A}^*_{h'_V}\,, \text{ with } \mathcal{A}_h = -\sqrt{2}g_V m_V e^{ih\bar{\varphi}}d_h(\bar{\theta})\,, \tag{A.8}$$

where d_h are Wigner-d functions, $\bar{\theta}$ and $\bar{\varphi}$ are the angles of the fermion or anti-fermion of $+1/2$ helicity produced in the decay in the rest frame of each boson. Once again, they are denoted with a bar to distinguish them from the observable quantities. See ref. [5] for details and for the explicit expression of the g_V couplings. Notice that the coupling for the Z boson decay depends on the chirality of the final-state leptons, denoted as χ in eq. (A.2).

We saw in section 1 that Monte Carlo generators work by sampling in a space of latent variables ξ, which are later projected into the space of the observable variables x. It is instructive to illustrate these generic considerations in the concrete case at hand, starting from listing the latent-space variables and next discussing their relation with the observables. The partonic cross section A.5 depends on the following latent variables

$$\xi = \{Q,\, \chi,\, \bar{s},\, \bar{\Theta},\, \bar{\theta}_W,\, \bar{\varphi}_W,\, \bar{\theta}_Z,\, \bar{\varphi}_Z\,, y\}\,. \tag{A.9}$$

Notice that the list does not include the total ZW transverse momentum $p_{T,\mathrm{ZW}}$, which vanishes exactly at tree level, and the flavor of the leptons produced in the decays, $\ell_{W,Z}$. The latter variables are excluded because electrons and muons behave identically in our approximation. The latent variable vector also includes the rapidity, y, of the partonic system center of mass frame. The partonic cross section A.5 is boost-invariant and hence independent of y. However, the rapidity is needed for the complete characterization of the kinematics of the event.

The electric charge of the W boson, Q, is the only variable in eq. (A.9) that can be measured experimentally, as the sum of the lepton charges. The chirality of the leptons from the Z boson decay, χ, is not observable. All the kinematical variables including $\bar{s}$ and $\bar{\Theta}$ depend on the momentum of the neutrino, which is not measured directly. Actually, even a direct measurement of the neutrino momentum would not enable the complete determination of all the variables. In particular, the Z boson angle $\bar{\Theta}$ must be measured relative to the direction of the incoming quark, which cannot be determined from the kinematics of the final-state particles. We circumvent this issue with the ideal generator by enforcing that the quark is always coming from the same direction. For the more realistic simulations at NLO, we instead compute the scattering angle assuming that the incoming quark is more energetic than the anti-quark. This is a reasonable approximation given the shape of the corresponding parton distribution functions. The approximation enables us to determine the direction of motion of the quark as the direction of the boost of the ZW system.

The difficulty in relating the latent-space variables to the observables is immaterial for event generation. We can merely sample the ξ variables according to their distribution (A.5),[13] compute the momentum of the three leptons and obtain the observables

$$x = \{Q\,, s,\, \Theta,\, \theta_W,\, \varphi_W,\, \theta_Z,\, \varphi_Z\}\,, \tag{A.10}$$

[13] We duly take into account parton distribution functions, using the nCTEQ15 [54] set and the `ManeParse` [55] Mathematica implementation. See [5] for details.

by running on the Monte Carlo data the same reconstruction algorithm that we would employ on the real data. The reconstruction involves the determination of the momentum of the neutrino which we perform, as previously explained, by imposing the on-shell condition for the W boson. A random choice is performed between the two solutions. The corresponding binary random variable should be regarded as an additional component of the latent-space vector (A.9).

Unlike in ref. [5], we are interested in Monte Carlo data sets that incorporate the dependence on the Wilson coefficients $c = (G_\varphi, G_W)$ by the technique of event reweighting. We thus generate unweighted events at the SM point $c = 0$ and assign to each of them a weight

$$w_{\rm e}(c) = w_{\rm e}(0)\frac{d\sigma(\xi_{\rm e},\, c)}{d\sigma(\xi_{\rm e},\, 0)}\,, \tag{A.11}$$

where $w_{\rm e}(0)$ is equal to the total cross section — as obtained from the Monte Carlo integration — divided by the number of events in the data set. The determination of the weights is trivial, owing to the knowledge of the value assumed by the latent variables for each event, $\xi_{\rm e}$, and to the analytic knowledge of the cross section A.5 in the latent space. On the other hand, the observable variables x do not contain enough information to compute the weights. In particular, this means that the weights cannot be computed on real data points. Notice $d\sigma(\xi,\, c)$, and in turn $w_{\rm e}(c)$ — is a quadratic polynomial of c. The coefficients of this polynomial are stored in the event file, providing the complete representation of the c-dependent weight function $w_{\rm e}(c)$ for each event.

The existence of a complicated relation between the latent and the observable variables is immaterial for event generation and for event reweighting, but it prevents in general the analytical calculation of the differential cross section, $d\sigma(x; c)$, in the space of observables. In turn, this prevents the determination of the cross section ratio $r(x; c)$. However, the relation between latent and observable variables is in fact simple within the assumptions that underpin the ideal description of the ZW process. In the narrow width approximation, the W particle is exactly on-shell, therefore the true momentum of the neutrino obeys the on-shell condition exactly. One of the two solutions to the on-shell equation — call them $p_\nu^{(a)}$ and $p_\nu^{(b)}$ — is thus exactly equal to the true momentum. Furthermore, it was noticed in ref. [46] that a very simple relation exists between the two solutions in the massless limit $m_W \to 0$. The reconstructed neutrino momenta approach each other in the limit, $p_\nu^{(a)} = p_\nu^{(b)}$, but they become collinear to the momentum of the lepton produced in the W decay. The W decay angles determined on the two solutions depend on how the collinear limit is approached, and they are not identical. One finds that

$$\varphi_W^{(a)} = \pi - \varphi_W^{(b)}\,, \tag{A.12}$$

while the two determinations of the polar angle are instead equal, $\theta_W^{(a)} = \theta_W^{(b)}$. All the other kinematical variables are identical on the two solutions because they both reconstruct the correct W boson momentum $p_W = p_\nu + p_\ell$. The reconstruction of the observable variables (A.10), starting from the latent variables (A.9), thus simply amounts to identify s, Θ, θ_W, θ_Z and φ_Z with their barred version and to assign

$$\varphi_W = \bar\varphi_W \quad\text{or}\quad \varphi_W = \pi - \bar\varphi_W\,, \tag{A.13}$$

at random with equal probability.

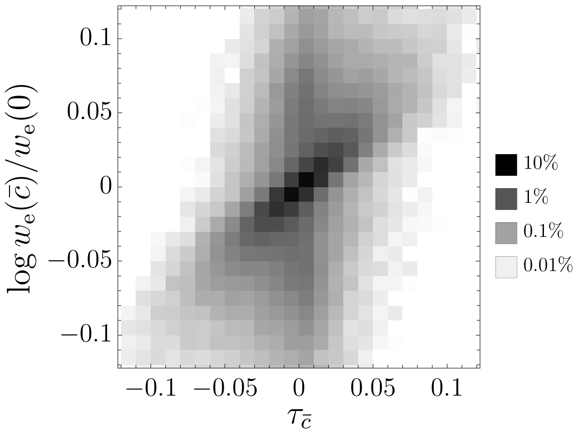


Figure 13. Density histogram of the correlation, on ideal Monte Carlo data, between the true distribution log-ratio $\tau(x_\mathrm{e}) = \log r(x_\mathrm{e}, \bar{c})$ and the logarithm of the weight ratio $w_\mathrm{e}(\bar{c})/w_\mathrm{e}(0)$. The $\bar{c}$ point has $G_\varphi = 0$ and $G_W = 10^{-2}\,\mathrm{TeV}^{-2}$.

These simplifications allow us to compute the differential cross section $d\sigma(x;c)$ easily, obtaining an exact analytical expression for the distribution ratio

$$r(x;c) = \frac{d\sigma(x;c)}{d\sigma(x;0)}\,. \tag{A.14}$$

As discussed in [5], one simply needs to average over the twofold ambiguity (A.13) in the reconstruction of φ_W, and to sum over the two possible chiralities $\chi = L, R$ of the Z boson decay. The knowledge of the exact distribution ratio will be used to validate the performances of our method on ideal data.

Given that $d\sigma(x;c)$ can be computed, our strategy for event generation appears unnecessarily complicated: we could have sampled directly in the x space and ignored latent variables altogether. Event reweighting could have been also easily implemented, leading to weights

$$\tilde{w}_\mathrm{e}(c) = w_\mathrm{e}(0)\frac{d\sigma(x_\mathrm{e};c)}{d\sigma(x_\mathrm{e};0)} = w_\mathrm{e}(0)\,r(x_\mathrm{e};c)\,. \tag{A.15}$$

Unlike the w_e weights that we will employ in the rest of this paper, defined by eq. (A.11), the $\tilde{w}_\mathrm{e}$ weights are uniquely determined by the observable variables x. Furthermore, they are proportional to the r-ratio computed on the event. However, this alternative approach to event generation that does not employ latent variables is not viable, because our purpose is to define an idealized but still realistic setup to validate our ability to learn r. The cross section in the x space cannot be computed for realistic simulators. The usage of latent variables is mandatory and we do not have access to the $\tilde{w}_\mathrm{e}$ weights in eq. (A.15), but only to the w_e weights (A.11) in the latent space. Using the knowledge of the $\tilde{w}_\mathrm{e}$ weights would define an unrealistically simple problem that is not suited for the validation of our methodology.

It is instructive to discuss in more detail why employing the $\tilde{w}_\mathrm{e}$ weights in place of the w_e weights would define an overly simple problem. Consider learning the distribution ratio at a specific point $c = \bar{c}$ in the parameter space, as described in section 2.1.1. We have shown that by employing a flexible set of trainable functions to model the classification function $f(x) \in (0,1)$ we obtain, after training with the loss function in eq. (2.1), a good approximation $\hat{r}(x;\bar{c})$ of the distribution ratio. Owing to eq. (2.3), after training

$$f(x) = \frac{1}{1+\hat{r}(x;\bar{c})}\,. \tag{A.16}$$

This result holds provided the training data offer a consistent representation of the x distribution, which is the case both if the reweighting is performed in the space of latent variables with the $w_{\rm e}$ weights, or if instead the $\tilde{w}_{\rm e}$ weights are used. However, in the latter case the training problem becomes trivial because the loss function in eq. (2.1) can be rewritten as

$$\ell[f(\cdot)] = \sum_{\rm e\in S}\left[w_{\rm e}(\bar{c}) + w_{\rm e}(0)\right]\left[f(x_{\rm e}) - \frac{1}{1+w_{\rm e}(\bar{c})/w_{\rm e}(0)}\right]^2 + \sum_{\rm e\in S}\frac{w_{\rm e}(c)\,w_{\rm e}(0)}{w_{\rm e}(c)+w_{\rm e}(0)}\,. \tag{A.17}$$

The second term is a constant, irrelevant for the minimization. Consider the first term, and suppose employing the $\tilde{w}_{\rm e}$ weights in place of the $w_{\rm e}$ weights. Since $\tilde{w}_{\rm e}(c)/\tilde{w}_{\rm e}(0) = r(x_{\rm e},\bar{c})$, by eq. (A.15), we see that the first term of the loss function is proportional to the square of the difference between the trainable function f and the target function $1/(1+r)$, evaluated on the training points. This is the typical form of the loss function one would employ for regression. In fact, eq. (A.17) shows that our learning problem reduces to a trivial problem of regression without noise, if the $\tilde{w}_{\rm e}$ weights were used for training. This would not offer a fair representation of the realistic problem.

In order to appreciate how far our actual problem is from a regression, we show in figure 13 a density histogram that displays the correlation between the true distribution ratio $r(x_{\rm e};\bar{c})$ and the ratio of the weights, $w_{\rm e}(\bar{c})/w_{\rm e}(0)$, on the generated events. More precisely, we plot $\tau = \log r$ versus the logarithm of the weights ratio. The $\bar{c}$ point used for illustration has $G_\varphi = 0$ and $G_W = 10^{-2}\,\text{TeV}^{-2}$. We observe a loose correlation between the two quantities. Our problem is thus very far from a regression. It could still be described as a regression, but one with large and non-Gaussian noise. We can compare figure 13 with the central and right panels of figure 1 in section 3.1, which displays the much tighter correlation between the true τ and the reconstructed $\hat{\tau} = \log\hat{r}$, obtained of course from training with the $w_{\rm e}$ weights and not with the $\tilde{w}_{\rm e}$ weights.

A.2 NLO generator

Our NLO event generator is based on the MadGraph_aMC@NLO [38] software suite, version 2.9.3, using Pythia 8.245 [56, 57] for showering. We generate the complete $2 \to 4$ processes

$$pp \to \ell^+\ell^+\ell^-\nu_\ell \;\;\text{and}\;\; pp \to \ell^-\ell^-\ell^+\bar{\nu}_\ell\,, \tag{A.18}$$

with $\ell = e, \mu$. Loose generation-level cuts are imposed to avoid singularities. The generated events are passed through Delphes [52] for clustering. The Delphes "GenJet" objects, which do no include detector effects, are used for the analysis. Clustering is performed with the anti-k_T algorithm with cone radius $\Delta R = 0.1$. If leptons and photons carry more than 90% of the GenJet cluster, the cluster is labeled as a lepton with the same flavor of the most energetic lepton. Otherwise, it is labeled as a QCD jet.

We impose the following acceptance cuts on the reconstructed leptons

$$p_T^\ell > 25 \text{ GeV}\,, \quad \eta^\ell < 2.5\,, \tag{A.19}$$

and a minimum separation $\Delta R > 0.4$ between the lepton and the jets. Only events with exactly three leptons that satisfy these requirements — two of which with opposite charges but the same flavor — are kept for the analysis. We further ask for a minimum missing energy $\not{E}_T > 30$ GeV.

The leptons coming from the Z boson have same flavor and opposite charge. If more than one lepton pair exists with these properties — because all three leptons have the same flavor — the pair with invariant mass closest to the mass of the Z boson is chosen. The other lepton is assumed to be coming from a W boson with the same charge as the lepton. After the Z and W decay products are identified, we compute the invariant mass m_{Z^*} of the virtual Z, the transverse mass m_{T,W^*} of the W boson and the transverse momenta of the Z and of the W, and we impose the selection cuts

$$70 \text{ GeV} < m_{Z^*} < 110 \text{ GeV}\,, \quad m_{T,W^*} < 90 \text{ GeV}\,, \quad \min[p_{T,W}, p_{T,Z}] > 300 \text{ GeV}\,. \tag{A.20}$$

We finally reconstruct the neutrino momentum as previously explained and compute the vector of observables

$$x = \{Q,\, \ell_Z,\, \ell_W\,,\, s,\, \Theta,\, \theta_W,\, \varphi_W,\, \theta_Z,\, \varphi_Z,\, p_{T,\mathrm{ZW}}\}\,. \tag{A.21}$$

The events are generated according to the SM hypothesis $c = 0$ and are then reweighted. We use the automated reweighting at NLO accuracy that is available in MadGraph_aMC@NLO, with the implementation of the EFT operators (A.4) that is provided in the SMEFTatNLO model [39]. The coefficients of the quadratic polynomial that defines $w_\mathrm{e}(c)$ are computed and stored in the event file.

In comparison with the ideal data in eq. (A.10), more observables are considered at NLO (A.21). The flavor $\ell_{Z,W}$ of the leptons are included because the NLO generator accounts for the effect of QED showering, through Pythia. The electron and the muon shower differently, hence their distributions are not identical unlike in ideal data. These additional discrete labels, like the charge Q, will be given as input to the neural network. NLO data also include the total transverse momentum of the ZW system, $p_{T,\mathrm{ZW}}$, which emerges from the real QCD radiation.

References

[1] J. Brehmer, K. Cranmer, G. Louppe and J. Pavez, *Constraining Effective Field Theories with Machine Learning*, *Phys. Rev. Lett.* **121** (2018) 111801 [arXiv:1805.00013] [INSPIRE].

[2] J. Brehmer, K. Cranmer, G. Louppe and J. Pavez, *A Guide to Constraining Effective Field Theories with Machine Learning*, *Phys. Rev. D* **98** (2018) 052004 [arXiv:1805.00020] [INSPIRE].

[3] J. Brehmer, G. Louppe, J. Pavez and K. Cranmer, *Mining gold from implicit models to improve likelihood-free inference*, *Proc. Nat. Acad. Sci.* **117** (2020) 5242 [arXiv:1805.12244] [INSPIRE].

[4] J. Brehmer, F. Kling, I. Espejo and K. Cranmer, *MadMiner: Machine learning-based inference for particle physics*, *Comput. Softw. Big Sci.* **4** (2020) 3 [arXiv:1907.10621] [INSPIRE].

[5] S. Chen, A. Glioti, G. Panico and A. Wulzer, *Parametrized classifiers for optimal EFT sensitivity*, *JHEP* **05** (2021) 247 [arXiv:2007.10356] [INSPIRE].

[6] S. Chatterjee, S. Rohshap, R. Schöfbeck and D. Schwarz, *Learning the EFT likelihood with tree boosting*, arXiv:2205.12976 [INSPIRE].

[7] S. Chatterjee et al., *Tree boosting for learning EFT parameters*, *Comput. Phys. Commun.* **277** (2022) 108385 [arXiv:2107.10859] [INSPIRE].

[8] E. Arganda et al., *A method for approximating optimal statistical significances with machine-learned likelihoods*, *Eur. Phys. J. C* **82** (2022) 993 [arXiv:2205.05952] [INSPIRE].

[9] E. Arganda, A.D. Perez, M. de los Rios and R.M. Sandá Seoane, *Machine-learned exclusion limits without binning*, *Eur. Phys. J. C* **83** (2023) 1158 [arXiv:2211.04806] [INSPIRE].

[10] R. Gomez Ambrosio et al., *Unbinned multivariate observables for global SMEFT analyses from machine learning*, *JHEP* **03** (2023) 033 [arXiv:2211.02058] [INSPIRE].

[11] K. Kong, K.T. Matchev, S. Mrenna and P. Shyamsundar, *New Machine Learning Techniques for Simulation-Based Inference: InferoStatic Nets, Kernel Score Estimation, and Kernel Likelihood Ratio Estimation*, arXiv:2210.01680 [INSPIRE].

[12] J. Hollingsworth and D. Whiteson, *Resonance Searches with Machine Learned Likelihood Ratios*, arXiv:2002.04699 [INSPIRE].

[13] S. Rizvi, M. Pettee and B. Nachman, *Learning likelihood ratios with neural network classifiers*, *JHEP* **02** (2024) 136 [arXiv:2305.10500] [INSPIRE].

[14] K. Kondo, *Dynamical Likelihood Method for Reconstruction of Events With Missing Momentum. 1: Method and Toy Models*, *J. Phys. Soc. Jap.* **57** (1988) 4126 [INSPIRE].

[15] P. Artoisenet, V. Lemaitre, F. Maltoni and O. Mattelaer, *Automation of the matrix element reweighting method*, *JHEP* **12** (2010) 068 [arXiv:1007.3300] [INSPIRE].

[16] F. Fiedler, A. Grohsjean, P. Haefner and P. Schieferdecker, *The Matrix Element Method and its Application in Measurements of the Top Quark Mass*, *Nucl. Instrum. Meth. A* **624** (2010) 203 [arXiv:1003.1316] [INSPIRE].

[17] T. Martini and P. Uwer, *Extending the Matrix Element Method beyond the Born approximation: Calculating event weights at next-to-leading order accuracy*, *JHEP* **09** (2015) 083 [arXiv:1506.08798] [INSPIRE].

[18] T. Martini and P. Uwer, *The Matrix Element Method at next-to-leading order QCD for hadronic collisions: Single top-quark production at the LHC as an example application*, *JHEP* **05** (2018) 141 [arXiv:1712.04527] [INSPIRE].

[19] J.M. Campbell, W.T. Giele and C. Williams, *The Matrix Element Method at Next-to-Leading Order*, *JHEP* **11** (2012) 043 [arXiv:1204.4424] [INSPIRE].

[20] S. Prestel and M. Spannowsky, *HYTREES: Combining Matrix Elements and Parton Shower for Hypothesis Testing*, *Eur. Phys. J. C* **79** (2019) 546 [arXiv:1901.11035] [INSPIRE].

[21] D. Atwood and A. Soni, *Analysis for magnetic moment and electric dipole moment form-factors of the top quark via* $e^+e^- \to t\bar{t}$, *Phys. Rev. D* **45** (1992) 2405 [INSPIRE].

[22] M. Diehl and O. Nachtmann, *Optimal observables for the measurement of three gauge boson couplings in* $e^+e^- \to W^+W^-$, *Z. Phys. C* **62** (1994) 397 [INSPIRE].

[23] I. Dunietz et al., *How to extract CP violating asymmetries from angular correlations*, *Phys. Rev. D* **43** (1991) 2193 [INSPIRE].

[24] A.S. Dighe, I. Dunietz and R. Fleischer, *Extracting CKM phases and* $B_s - \bar{B}_s$ *mixing parameters from angular distributions of nonleptonic B decays*, *Eur. Phys. J. C* **6** (1999) 647 [hep-ph/9804253] [INSPIRE].

[25] G. Durieux and Y. Grossman, *Probing CP violation systematically in differential distributions*, *Phys. Rev. D* **92** (2015) 076013 [arXiv:1508.03054] [INSPIRE].

[26] G. Durieux, M. Perelló, M. Vos and C. Zhang, *Global and optimal probes for the top-quark effective field theory at future lepton colliders*, *JHEP* **10** (2018) 168 [arXiv:1807.02121] [INSPIRE].

[27] J. Pretz and F. Müller, *Extraction of Azimuthal Asymmetries using Optimal Observables*, *Eur. Phys. J. C* **79** (2019) 47 [arXiv:1811.09452] [INSPIRE].

[28] B. Bortolato, J.F. Kamenik, N. Košnik and A. Smolkovič, *Optimized probes of* CP *-odd effects in the* $t\bar{t}h$ *process at hadron colliders*, *Nucl. Phys. B* **964** (2021) 115328 [arXiv:2006.13110] [INSPIRE].

[29] D.A. Faroughy, J.F. Kamenik, N. Košnik and A. Smolkovič, *Probing the* CP *nature of the top quark Yukawa at hadron colliders*, *JHEP* **02** (2020) 085 [arXiv:1909.00007] [INSPIRE].

[30] Y. Gao et al., *Spin Determination of Single-Produced Resonances at Hadron Colliders*, *Phys. Rev. D* **81** (2010) 075022 [arXiv:1001.3396] [INSPIRE].

[31] J.S. Gainer et al., *Exploring Theory Space with Monte Carlo Reweighting*, *JHEP* **10** (2014) 078 [arXiv:1404.7129] [INSPIRE].

[32] S. Frixione and B.R. Webber, *Matching NLO QCD computations and parton shower simulations*, *JHEP* **06** (2002) 029 [hep-ph/0204244] [INSPIRE].

[33] S. Alioli, P. Nason, C. Oleari and E. Re, *A general framework for implementing NLO calculations in shower Monte Carlo programs: the POWHEG BOX*, *JHEP* **06** (2010) 043 [arXiv:1002.2581] [INSPIRE].

[34] R. Frederix et al., *Higgs pair production at the LHC with NLO and parton-shower effects*, *Phys. Lett. B* **732** (2014) 142 [arXiv:1401.7340] [INSPIRE].

[35] K. Arnold, J. Bella, J. Bellm, J. Bozzi, M. Brieg, F. Campanario et al., *REPOLO: REweighting POwheg events at Leading Order* , https://www.itp.kit.edu/vbfnlo/wiki/lib/exe/fetch.php?media=documentation:repolo_1.0.pdf.

[36] O. Mattelaer, *On the maximal use of Monte Carlo samples: re-weighting events at NLO accuracy*, *Eur. Phys. J. C* **76** (2016) 674 [arXiv:1607.00763] [INSPIRE].

[37] A. Andreassen and B. Nachman, *Neural Networks for Full Phase-space Reweighting and Parameter Tuning*, *Phys. Rev. D* **101** (2020) 091901 [arXiv:1907.08209] [INSPIRE].

[38] J. Alwall et al., *The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations*, *JHEP* **07** (2014) 079 [arXiv:1405.0301] [INSPIRE].

[39] C. Degrande et al., *Automated one-loop computations in the standard model effective field theory*, *Phys. Rev. D* **103** (2021) 096024 [arXiv:2008.11743] [INSPIRE].

[40] G. Cowan, K. Cranmer, E. Gross and O. Vitells, *Asymptotic formulae for likelihood-based tests of new physics*, *Eur. Phys. J. C* **71** (2011) 1554 [*Erratum ibid.* **73** (2013) 2501] [arXiv:1007.1727] [INSPIRE].

[41] J. Neyman and E.S. Pearson, *On the Problem of the Most Efficient Tests of Statistical Hypotheses*, *Phil. Trans. Roy. Soc. Lond. A* **231** (1933) 289 [INSPIRE].

[42] A. Falkowski, M. Gonzalez-Alonso, A. Greljo and D. Marzocca, *Global constraints on anomalous triple gauge couplings in effective field theory approach*, *Phys. Rev. Lett.* **116** (2016) 011801 [arXiv:1508.00581] [INSPIRE].

[43] D.R. Green, P. Meade and M.-A. Pleier, *Multiboson interactions at the LHC*, *Rev. Mod. Phys.* **89** (2017) 035008 [arXiv:1610.07572] [INSPIRE].

[44] A. Butter et al., *The Gauge-Higgs Legacy of the LHC Run I*, *JHEP* **07** (2016) 152 [arXiv:1604.03105] [INSPIRE].

[45] R. Franceschini et al., *Electroweak Precision Tests in High-Energy Diboson Processes*, *JHEP* **02** (2018) 111 [arXiv:1712.01310] [INSPIRE].

[46] G. Panico, F. Riva and A. Wulzer, *Diboson interference resurrection*, *Phys. Lett. B* **776** (2018) 473 [arXiv:1708.07823] [INSPIRE].

[47] A. Azatov, J. Elias-Miro, Y. Reyimuaji and E. Venturini, *Novel measurements of anomalous triple gauge couplings for the LHC*, *JHEP* **10** (2017) 027 [arXiv:1707.08060] [INSPIRE].

[48] A. Azatov, D. Barducci and E. Venturini, *Precision diboson measurements at hadron colliders*, *JHEP* **04** (2019) 075 [arXiv:1901.04821] [INSPIRE].

[49] J. Baglio, S. Dawson and S. Homiller, *QCD corrections in Standard Model EFT fits to WZ and WW production*, *Phys. Rev. D* **100** (2019) 113010 [arXiv:1909.11576] [INSPIRE].

[50] A. Paszke et al., *PyTorch: An Imperative Style, High-Performance Deep Learning Library*, arXiv:1912.01703 [INSPIRE].

[51] D.P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, arXiv:1412.6980 [INSPIRE].

[52] DELPHES 3 collaboration, *DELPHES 3, A modular framework for fast simulation of a generic collider experiment*, *JHEP* **02** (2014) 057 [arXiv:1307.6346] [INSPIRE].

[53] R.T. d'Agnolo et al., *Learning new physics from an imperfect machine*, *Eur. Phys. J. C* **82** (2022) 275 [arXiv:2111.13633] [INSPIRE].

[54] A. Kusina et al., *nCTEQ15 — Global analysis of nuclear parton distributions with uncertainties*, *PoS* **DIS2015** (2015) 041 [arXiv:1509.01801] [INSPIRE].

[55] D.B. Clark, E. Godat and F.I. Olness, *ManeParse: A Mathematica reader for Parton Distribution Functions*, *Comput. Phys. Commun.* **216** (2017) 126 [arXiv:1605.08012] [INSPIRE].

[56] T. Sjostrand, S. Mrenna and P.Z. Skands, *PYTHIA 6.4 Physics and Manual*, *JHEP* **05** (2006) 026 [hep-ph/0603175] [INSPIRE].

[57] T. Sjostrand, S. Mrenna and P.Z. Skands, *A Brief Introduction to PYTHIA 8.1*, *Comput. Phys. Commun.* **178** (2008) 852 [arXiv:0710.3820] [INSPIRE].