IOP Publishing Mach. Learn.: Sci. Technol. 6 (2025) 015007 https://doi.org/10.1088/2632-2153/ad9fd1

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
16 July 2024

REVISED
29 November 2024

ACCEPTED FOR PUBLICATION
15 December 2024

PUBLISHED
13 January 2025

PAPER

Refinable modeling for unbinned SMEFT analyses

Robert Schöfbeck

Institute for High Energy Physics, Austrian Academy of Sciences, Dominikanerbastei 16, Vienna 1010, Austria
Technische Universität Wien, Karlsplatz 13, Vienna 1040, Austria

E-mail: robert.schoefbeck@oeaw.ac.at

Keywords: systematic uncertainties, effective field theory, tree algorithms, boosting

Abstract

We present methods to estimate systematic uncertainties in unbinned large hadron collider (LHC) data analyses, focusing on constraining Wilson coefficients in the standard model effective field theory (SMEFT). Our approach also applies to broader parametric models of non-resonant phenomena beyond the standard model. By using machine-learned surrogates of the likelihood ratio, we extend well-established procedures from binned Poisson counting experiments to the unbinned case. This framework handles various theoretical, modeling, and experimental uncertainties, laying the foundation for future unbinned analyses at the LHC. We also introduce a tree-boosting algorithm that learns precise parametrizations of systematic effects, providing a robust, flexible alternative to neural networks for modeling systematics. We demonstrate this approach with an SMEFT analysis of highly energetic top quark pair production in proton–proton collisions.

Contents

1. Introduction

The large hadron collider (LHC) generates vast amounts of data from particle decays in high-energy interactions, offering a unique opportunity to explore fundamental physics. Recent advances in machine learning (ML) provide powerful tools not only for reconstruction and object-tagging but also for novel analysis techniques. High-dimensional unbinned analyses, where dozens of features probe a large number of model parameters, are now feasible with machine-learned surrogates optimized for hypothesis testing.

In the theoretical domain, the lack of new resonance signals has led to the adoption of the standard model effective field theory (SMEFT) [1–5] as the main framework for describing phenomena below an assumed energy scale, conventionally set at $\Lambda_{\text{SMEFT}} = 1$ TeV. This framework extends the standard model (SM) Lagrangian with field monomials, where the Wilson coefficients serve as the parameters of interest (POIs). The SMEFT enables experimentalists to test a range of high-scale models without dealing with their fundamental parameters.

The SMEFT is organized by the mass dimension of operators, beginning with dimension six for relevant new physics scenarios at the LHC [6]. Since the lowest-order matrix-element (ME) modifications to Wilson coefficients are linear, cross-section deviations can be described by quadratic polynomials within the SMEFT's validity range [7]. This analytic structure supports simulation-based inference (SBI) methods that improve performance, especially when probing multiple Wilson coefficients simultaneously [8–19]. These methods offer statistically optimal observables at the detector level, with fast evaluation after an initial training stage.

Nevertheless, most current LHC measurements are straightforward Poisson counting experiments, partially because these reduce the computational demand. Large-scale computing infrastructure has become more accessible, but the application of unbinned techniques is still hampered by the absence of a comprehensive set of tools that bring decades of experience with treating systematic effects in binned Poisson measurements on par with the unbinned case. The available methodology for treating systematic uncertainties in unbinned SBI techniques, such as ML optimal observables, is sparse. While optimal ML observables have a sound footing in well-developed statistical methodology, the otherwise finely honed procedures for treating systematic uncertainties are rarely seen in this light. The present work aims to change that situation through a comprehensive statistical interpretation of procedures for treating systematic effects in SBI. We explain how the factorization of individual systematic effects facilitates the training of multi-variate parametrized regressors and how to address uncertainties in the normalization of processes. The most significant advantage of this approach is its stage-wise nature. Adding new processes or systematic uncertainties does not invalidate partially available training.

The conceptual cornerstone of the modeling of collider phenomena underpinning SBI relies on a hierarchical separation of processes by energy scale, starting from hypothetical UV phenomena and their SMEFT parametrization at Λ_{SMEFT}. The SMEFT Wilson coefficients, our POIs, are denoted by $\boldsymbol{\theta}$, and we aim at parameter inference through frequentist confidence intervals [21]. Those parameters are, therefore, not stochastic. However, we note that there is no conceptual limitation to Bayesian SMEFT analysis.

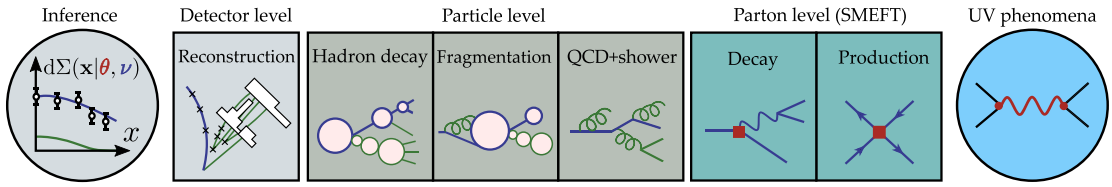

Figure 1. A hierarchy of scales governs our assumed understanding of physical phenomena and the modeling of collider physics. Staged event simulation proceeds in the 'forward mode' from right to left, while data analyses proceed in the reverse direction, aimed at constraining models of UV phenomena or the SMEFT POIs from the detector-level measurements. Reproduced from [20]. CC BY 4.0.

Figure 1 reflects this hierarchy, from right to left, by grouping unobserved (latent) variables and systematic effects at the parton, particle, and detector levels. This division balances sufficient detail with manageable notation complexity; more levels could be added but would obscure the core concepts. Collider event simulation mirrors this staging: at the parton level, ME generators sample SMEFT ME-squared terms. Key systematic uncertainties here include unphysical effects from technically unavoidable energy scales linked to the perturbative renormalization of fixed-order predictions and the factorization of collinear and infrared radiation. Uncertainties in the parton distribution functions (PDFs) cover our lack of knowledge in the composition of the pp initial state. At the scale of Λ_{QCD}, particle-level simulations handle parton shower (PS) effects, fragmentation, hadronization, decay, and underlying event modeling. Tuning these models introduces systematic uncertainties, and modeling generator differences as a 'two-point' systematic uncertainty can be effective. At the detector level, simulations include particle interactions with detector material, digitization, trigger logic, and event reconstruction.

Our approach provides a general method to obtain ML-based parametrizations of all these effects by grouping them into classes of systematic variations addressed individually. With these procedures in place, we can build unbinned models that can be iteratively refined. Adding new effects or background contributions does not invalidate surrogates trained on the initial model.

On the technical side, we fill a gap in the methodology by developing a tree-boosting algorithm that can learn arbitrarily accurate parametrizations of systematic effects. This is done by extending tree algorithms to produce regressors that are parametric in externally provided data; in our case, the nuisance parameters ($\boldsymbol{\nu}$) linked to systematic effects. The resulting 'Boosted Parametric Tree' (BPT) offers a robust and flexible alternative to neural networks for this purpose, with the training procedure fully grounded in unbinned model building. Therefore, it enables a full understanding of tree-boosting within the context of unbinned hypothesis tests. The terminal nodes of the trees act as measurement bins, allowing for an analytic dependence on nuisance parameters similar to the binned case. The BPT thus learns an expressive surrogate for differential cross-section ratios (DCRs), accommodating a potentially high-dimensional set of model parameters.

We use unbinned SMEFT analyses as our motivating case, assuming that the dependence on the POIs is learned by an algorithm from the literature [8–17]. We demonstrate our tools for modeling systematic effects through a semi-realistic SMEFT case study in top quark pair production. However, this methodology is broadly applicable and could enhance the inference of any SM parameter with a non-resonant impact on collider data. In addition to extracting parameters like α_s, electro-weak precision observables, or $\sin^2\boldsymbol{\theta}_{\mathrm{W}}$, inclusive cross-section measurements could benefit from an unbinned treatment of the signal process.

The rest of the paper is structured as follows. We discuss the relation to existing works for unbinned SMEFT analyses in section 2. The statistical setup for unbinned hypothesis tests is provided in section 3, and we use it to outline the key concepts of refinable modeling. A comprehensive review of the statistical interpretation of event simulation, suitable for developing ML tools, is given in section 4. This part also defines the terminology for section 5, which explains how to train generic regressors for suitable parametrizations of the various model-parameter dependencies and introduces the BPT algorithm. In section 6, we use the enw tools as building blocks for constructing refinable unbinned models. Those are applied in section 7, where we demonstrate the application of the procedures for SMEFT analyses of top quark pair production in the two-lepton channel. We provide conclusions in section 8.

2. Relation to other works

Several approaches in the literature suggest ML optimal test statistics, and we incorporate aspects of the statistical methodology and ML techniques.

The **ML4EFT** framework [8] advocates unbinned SMEFT hypothesis tests using a similar statistical setting as this work and presents sensitivity studies obtained without detector simulation or systematic uncertainties. The present work takes the next step, focussing on treating systematic uncertainties, and provides the tools for capitalizing on simulated data sets, encapsulating the systematic effects from all stages of LHC event simulation. A less significant difference is that SMEFT effects in [8] are learned by using networks, while we use the following tree-based method for this purpose.

The **Boosted information tree** (**BIT**) [9, 10] is a tree-based algorithm for learning SMEFT effects. In this work, we adopt it for the SMEFT signal modeling and extend it toward predicting the full positive quadratic SMEFT polynomial. The BIT algorithm learns the quadratic SMEFT polynomial using the same statistical foundation as the present work. The 'Parametric Regression Tree' in section 5.4 has a different goal, but the technical implementation and the statistical interpretation of the boosting are closely related.

In **Parametrized classifiers for optimal EFT sensitivity** [11], the authors develop a neural-network-based approach for learning optimal SMEFT classifiers up to next-to-leading (NLO) perturbative accuracy. Chen *et al* [12] provides a reweighting-based extension. Our approach to learning the SMEFT signal dependence is a tree-based alternative, and our focus in this work is on systematic uncertainties. The idea of learning generic coefficient functions for parametric regressors, as discussed in section 5, is partly motivated by the corresponding SMEFT construction in [11].

The authors of the **MadMiner** framework [13–18] developed the understanding of event simulation for likelihood-free inference that is also an essential basis for this work. On the technological side, MadMiner provides various techniques for general parameter inference and, specifically, also for unbinned SMEFT analyses. Beyond these motivating examples, MadMiner arguably established the SBI methods as a subfield in high-energy physics, to which the present work contributes. While MadMiner can also model systematic uncertainties, we use a more general and incrementally refinable statistical model for this purpose.

The authors of **Learning new physics from an imperfect machine** [22] use an entirely different (neural-network-based) model of the phenomena beyond the SM, which is at variance with the SMEFT case presented here. Nevertheless, the statistical setup (section 3), in particular, the definition of the nuisance parameters and the parametrization of systematic effects, are similar to this work.

In the **Inferno** approach [23], a non-linear summary statistic is constructed by minimizing inference-motivated losses via stochastic gradient descent. The algorithm uses Fisher's information on the POIs and accounts for nuisance parameters, but it is not specific to SMEFT. In [24], the method is used to reduce the systematic uncertainties in the measurement of the top quark pair production in the τ+jets channel.

3. Unbinned likelihood ratio tests

Given a data set $\mathcal{D}$, confidence level (CL) intervals for the POIs $\boldsymbol{\theta}$ are determined from the profiled likelihood ratio test statistic. In this section, we relate it to quantities that a machine can learn. Splitting the data in a primary set $\mathcal{D}$ and an auxiliary set $\mathcal{A}$, we have

$$q_{\boldsymbol{\theta}}(\mathcal{D}) = -2\log\frac{\max_{\boldsymbol{\nu}} \; L(\mathcal{D},\mathcal{A}|\boldsymbol{\theta},\boldsymbol{\nu})}{\max_{\boldsymbol{\nu},\boldsymbol{\theta}} L(\mathcal{D},\mathcal{A}|\boldsymbol{\theta},\boldsymbol{\nu})} = -2\log\frac{L(\mathcal{D},\mathcal{A}|\boldsymbol{\theta},\hat{\boldsymbol{\nu}}_{\boldsymbol{\theta}})}{L\left(\mathcal{D},\mathcal{A}|\hat{\boldsymbol{\theta}},\hat{\boldsymbol{\nu}}\right)}. \tag{1}$$

The auxiliary data set has components to constrain systematic uncertainties, such as in the integrated luminosity or the jet energy scale calibration. The maximum-likelihood estimate (MLE) of the nuisance parameters for a given $\boldsymbol{\theta}$ is $\hat{\boldsymbol{\nu}}_{\boldsymbol{\theta}}$, while $(\hat{\boldsymbol{\theta}},\hat{\boldsymbol{\nu}})$ represents the MLE or all model parameters simultaenously.

By design, $q_{\boldsymbol{\theta}}(\mathcal{D})$ is always non-negative for any $\boldsymbol{\theta}$, with larger values indicating greater incompatibility between the model defined by $\boldsymbol{\theta}$ and the data $\mathcal{D}$. Without nuisance parameters, the Neyman–Pearson lemma states that a hypothesis test of fixed size α based on $q_{\boldsymbol{\theta}}(\mathcal{D})$ has maximum power, meaning it is most likely to correctly reject the null hypothesis when the alternative is true [25].

Technically, $\mathcal{A}$ should be an argument of $q_{\boldsymbol{\theta}}(\mathcal{D})$, but we omit it in the notation, as an analytic approximation of the corresponding likelihood factor, described below, will capture all its effects. We assume that $\mathcal{A}$ and $\mathcal{D}$ do not overlap and that SMEFT effects, parametrized by $\boldsymbol{\theta}$, are negligible in $\mathcal{A}$. Under this assumption, the auxiliary data set produces a multiplicative term $L(\mathcal{A}|\boldsymbol{\nu})$ in the likelihood,

$$L(\mathcal{D},\mathcal{A}|\boldsymbol{\nu},\boldsymbol{\theta}) = L(\mathcal{D}|\boldsymbol{\nu},\boldsymbol{\theta})\,L(\mathcal{A}|\boldsymbol{\nu}). \tag{2}$$

According to Wilks' theorem [26], if $\mathcal{D}$ is distributed under $\boldsymbol{\theta}$, then $q_{\boldsymbol{\theta}}(\mathcal{D})$ is asymptotically distributed as a $\chi^2_{N_{\boldsymbol{\theta}}}$ distribution, where $N_{\boldsymbol{\theta}}$ is the number of independent degrees of freedom in $\boldsymbol{\theta}$. This asymptotic distribution is independent of the nuisance parameters, which simplifies the limit-setting procedure and is a

primary reason why LHC data analyses commonly use the profiled likelihood ratio test statistic. The CLs at a CL of, e.g. $1-\alpha = 95\%$, are given by solving

$$q_{\boldsymbol{\theta}} = F^{-1}_{\chi^2_{N_{\boldsymbol{\theta}}}}(1-\alpha), \tag{3}$$

where $F_{\chi^2_{N_{\boldsymbol{\theta}}}}$ is the cumulative distribution function of the $\chi^2_{N_{\boldsymbol{\theta}}}$ distribution.

To obtain confidence intervals with some POIs profiled, these parameters are treated as nuisance parameters in $q_{\boldsymbol{\theta}}$, reducing $N_{\boldsymbol{\theta}}$ accordingly. However, large quadratic terms in the SMEFT expansion can invalidate Wilks' theorem [27]. In such cases, the distribution must be determined by other means, such as toy experiments, to ensure the desired CL.

The likelihood functions in equation (1) are 'extended' likelihoods: a Poisson-distributed counting variable describes the random fluctuation in the total number of observed events. The remaining discriminating information is encoded in the fiducial detector-level probability density function (pdf) $p(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\nu})$, which relates to the fiducial differential cross-section as

$$\mathrm{d}\Sigma(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\nu}) = \sigma(\boldsymbol{\theta},\boldsymbol{\nu})\, p(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\nu})\, \mathrm{d}\boldsymbol{x}. \tag{4}$$

We denote the inclusive fiducial cross-section by $\sigma(\boldsymbol{\theta},\boldsymbol{\nu})$. In general, $\mathrm{d}\Sigma(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\nu})$ includes contributions from multiple processes. With the integrated luminosity $\mathcal{L}(\boldsymbol{\nu})$, subject to systematic uncertainties whose effects we parametrize by nuisance parameters, the likelihood to observe a data set $\mathcal{D}$ of size N can, in general, be written in terms of the differential cross-section as

$$L(\mathcal{D}|\boldsymbol{\theta},\boldsymbol{\nu}) = \mathrm{P}_{\mathcal{L}(\boldsymbol{\nu})\sigma(\boldsymbol{\theta},\boldsymbol{\nu})}(N)\prod_{i=1}^{N} p(\boldsymbol{x}_i|\boldsymbol{\theta},\boldsymbol{\nu}) = \mathrm{P}_{\mathcal{L}(\boldsymbol{\nu})\sigma(\boldsymbol{\theta},\boldsymbol{\nu})}(N)\prod_{i=1}^{N}\frac{1}{\sigma(\boldsymbol{\theta},\boldsymbol{\nu})}\frac{\mathrm{d}\Sigma(\boldsymbol{x}_i|\boldsymbol{\theta},\boldsymbol{\nu})}{\mathrm{d}\boldsymbol{x}}, \tag{5}$$

where P_λ denotes the Poisson distribution with mean λ. The extended log-likelihood ratio for two sets of model parameters becomes

$$\log\frac{L(\mathcal{D}|\boldsymbol{\theta}_1,\boldsymbol{\nu}_1)}{L(\mathcal{D}|\boldsymbol{\theta}_0,\boldsymbol{\nu}_0)} = -\mathcal{L}(\boldsymbol{\nu}_1)\sigma(\boldsymbol{\theta}_1,\boldsymbol{\nu}_1) + \mathcal{L}(\boldsymbol{\nu}_0)\sigma(\boldsymbol{\theta}_0,\boldsymbol{\nu}_0) + \sum_{i=1}^{N}\log\left(\frac{\mathcal{L}(\boldsymbol{\nu}_1)}{\mathcal{L}(\boldsymbol{\nu}_0)}\frac{\mathrm{d}\Sigma(\boldsymbol{x}_i|\boldsymbol{\theta}_1,\boldsymbol{\nu}_1)}{\mathrm{d}\Sigma(\boldsymbol{x}_i|\boldsymbol{\theta}_0,\boldsymbol{\nu}_0)}\right) \tag{6}$$

each of the K sources of systematic uncertainty is associated with a nuisance parameter ν_k, collectively denoted by $\boldsymbol{\nu}$. Systematic effects related to detector calibration, the measurement of the integrated luminosity, theoretical calculations, and more are modeled with these nuisance parameters. Measurements can constrain some of these uncertainties through the observed auxiliary data set $\mathcal{A}_0$, which is the specific instance of $\mathcal{A}$ found in real data and, therefore, not a random quantity [22]. We set $\boldsymbol{\nu} = \mathbf{0}$ to correspond to the maximum of $L(\mathcal{A}_0|\boldsymbol{\nu})$, so that by definition,

$$\max_{\boldsymbol{\nu}} L(\mathcal{A}_0|\boldsymbol{\nu}) = L(\mathcal{A}_0|\mathbf{0}) \tag{7}$$

the central value $\boldsymbol{\nu} = \mathbf{0}$ represents the best available calibrations across all modeling aspects before considering $\mathcal{D}$. The nuisance parameters then parametrize deviations from this hypothesis. Combined with $\boldsymbol{\theta} = \mathbf{0}$, this choice defines an SM reference hypothesis with likelihood

$$L(\mathcal{D},\mathcal{A}|\mathrm{SM}) \equiv L(\mathcal{D}|\mathbf{0},\mathbf{0})\, L(\mathcal{A}|\mathbf{0}) \tag{8}$$

which describes the SM without any SMEFT effects and includes the best available calibrations. We can parametrize the systematic effects to make the ν_k as uncorrelated as possible and scale them so that the auxiliary log-likelihood ratio becomes a simple analytic expression in terms of $\boldsymbol{\nu}$. In the Gaussian approximation, this is achieved by diagonalizing the Hessian of the auxiliary likelihood function at $\boldsymbol{\nu} = \mathbf{0}$, resulting in a penalty of the form

$$-2\log\frac{L(\mathcal{A}|\boldsymbol{\nu})}{L(\mathcal{A}|\mathbf{0})} = \sum_{k=1}^{K}\nu_k^2 \tag{9}$$

though generalizations to other probability distributions are possible.

If a specific nuisance parameter is only constrained by the primary data set and not by $\mathcal{A}_0$, it is excluded from the penalty term in equation (9) and referred to as 'floating'. While some uncertainties, such as those related to PDFs, are clearly interpretable in terms of SM parameters, this is not always the case. For example, uncertainties from renormalization or factorization scales address limitations in perturbative accuracy but

do not guarantee statistical coverage when these scales vary in the simulation. With this caveat in mind, we treat all systematic uncertainties heuristically in the same way.

We normalize the likelihoods in equation (1) by dividing both the numerator and denominator by the SM reference likelihood in equation (8). This yields

$$q_{\boldsymbol{\theta}}(\mathcal{D}) = \min_{\boldsymbol{\nu}} u(\mathcal{D}, \mathcal{A}|\boldsymbol{\nu}, \boldsymbol{\theta}) - \min_{\boldsymbol{\nu},\boldsymbol{\theta}} u(\mathcal{D}, \mathcal{A}|\boldsymbol{\nu}, \boldsymbol{\theta}), \tag{10}$$

where

$$-\frac{1}{2}u(\mathcal{D}, \mathcal{A}|\boldsymbol{\nu}, \boldsymbol{\theta}) = -\mathcal{L}(\boldsymbol{\nu})\,\sigma(\boldsymbol{\theta}, \boldsymbol{\nu}) + \mathcal{L}_0\,\sigma(\mathrm{SM}) + \sum_{i=1}^{N(\mathcal{D})} \log\left(\frac{\mathcal{L}(\boldsymbol{\nu})}{\mathcal{L}_0}\frac{\mathrm{d}\Sigma(\boldsymbol{x}_i|\boldsymbol{\theta}, \boldsymbol{\nu})}{\mathrm{d}\Sigma(\boldsymbol{x}_i|\mathrm{SM})}\right) - \frac{1}{2}\sum_{k=1}^{K}\nu_k^2. \tag{11}$$

And $\mathcal{L}_0$ denotes the central value of the auxiliary luminosity measurement.

The main drawback of the unbinned likelihood ratio test statistic in equation (11) is the need to evaluate the DCR, inclusive cross-section, and integrated luminosity as functions of the model parameters. While log-normal (multiplicative) nuisances effectively model the integrated luminosity dependence around the central value [28, 29], we also require estimates for the DCR and inclusive cross-section. Event generators cannot provide these estimates parametrically in terms of model parameters, as they operate in 'forward' mode through sequential stochastic processes. However, for inference, the minimization in equation (1) requires to evaluate the DCR for externally provided simulated or real events.

4. Simulation for inference

Although computationally intensive, robust predictions from Monte Carlo (MC) simulations are invaluable for modeling highly energetic scattering processes [30]. Our strategy is to provide the necessary parametric estimates by breaking down the primary task of modeling the DCR into smaller, manageable machine-learning tasks. The tasks are based on simulation and capitalize on the high quality of the MC methods. As illustrated in figure 2, we proceed in the backward mode, from left to right, and separate the training into distinct parton-level processes that are simulated separately. For each process, we learn parametrizations of systematic effects and POI dependencies with the help of efficiently generated 'synthetic' data sets that correspond to the systematic variations in the binned approach. Inheriting these procedures, we leverage the extensive experience from over a decade of binned LHC data analyses.

Training tasks for systematics can be divided into uncorrelated groups of nuisance parameters which then separately estimate these effects. A high granularity allows a gradual refinement of each aspect of the final model without invalidating unrelated tasks. In the following sections, we describe how the results of ML training tasks combine into a refinable surrogate model $\mathrm{d}\Sigma(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{\nu})$ that provides the needed DCR in equation (11).

4.1. Hierarchical data representations and staged event simulation

Systematic effects can modify predictions at any energy scale. As shown in figure 1, we broadly categorize these into parton level (p), particle level (ptl), detector reconstruction level (reco), and observed features derived from reconstruction, denoted by $\boldsymbol{x}$. By definition, the observables $\boldsymbol{x}$ are the quantities used in equations (4)–(11). All other quantities, including $\boldsymbol{z}_{\mathrm{reco}}$ (e.g. low-level hit patterns in sub-detectors accessible in real data), are denoted by $\boldsymbol{z}$.

This grouping strikes a balance between detail and notational simplicity. It also reflects the approximate separation of systematic effects from UV and SMEFT energy scales to those relevant to QCD and detector signals. This staging is flexible; additional stages can be introduced as long as an event representation can be defined.

Event simulation for LHC proton-proton (pp) collisions is divided among several computer codes, each addressing modeling at a specific energy scale with specialized techniques. At the parton level, ME generators like MG5_aMC@NLO [31, 32], SHERPA [33], and POWHEG [34–38] provide a sampling of purely perturbative ME-squared predictions for the hard-scatter interaction. The dynamics of fundamental particles from the hard scatter and subsequent decays of heavy SM particles are represented by $\boldsymbol{z}_{\mathrm{p}}$. The PS evolves $\boldsymbol{z}_{\mathrm{p}}$ down to energies where perturbative methods are no longer valid. Together with color reconnection, hadronization, decays of unstable hadrons, underlying event, particles from multiple-parton interactions, and both initial- and final-state radiation, it defines the particle level. This is simulated with general-purpose tools like PYTHIA [39] or HERWIG[40], which require tuning of phenomenological parameters to reliably describe data. The resulting particle-level event is represented by $\boldsymbol{z}_{\mathrm{ptl}}$. The parton and particle levels are latent and cannot be directly observed in real data.

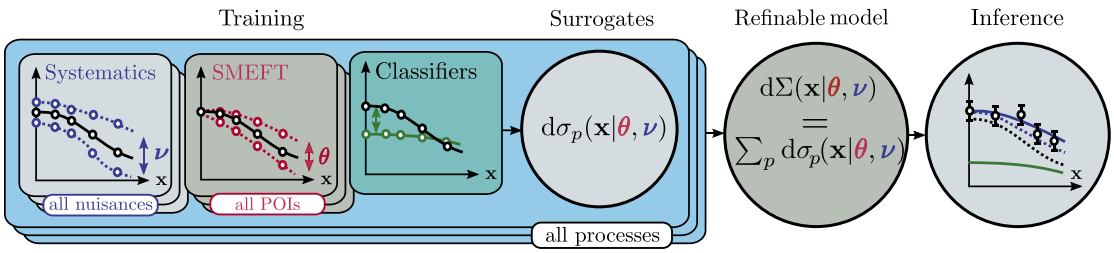

Figure 2. Overview of refinable modeling in SBI, proceeding in the backward mode, from left to right. For each process, we machine-learn the dependence on the SMEFT POIs and on the nuisance parameters and combine these elements into a refinable surrogate model for the global DCR in the measurement region, suitable for inference.

Particle-level events are processed with detector-specific simulation tools like GEANT [41], using conditions for each data-taking period and mixed with simulations of separate hard-scatter events to model pile-up. After simulating detector hits, event reconstruction proceeds to the particle candidate level, using, e.g. a version of the PARTICLEFLOW algorithm [42, 43]. Together with jet clustering, lepton identification, and disambiguation, this process is similar in both real and simulated data. The result is the reconstruction-level event, with features denoted by $\boldsymbol{z}_{\text{reco}}$. Simplified event reconstruction is available from, e.g. DELPHES [44].

Most data analyses derive high-level observables $\boldsymbol{x}$ from $\boldsymbol{z}_{\text{reco}}$. These observables capture all event features included in the hypothesis test. In principle, $\boldsymbol{x}$ can represent the entire reconstruction level, including the variable-length list of all reconstructed particle candidates in an event [45]. Such approaches have been used to constrain SMEFT effects [46]; here, we focus on high-level event features. Any real-data event features not already included in $\boldsymbol{x}$ belong to $\boldsymbol{z}_{\text{reco}}$ and are latent in the hypothesis test.

For bookkeeping, we group the nuisance parameters into $\boldsymbol{\nu}_{\text{p}}$, $\boldsymbol{\nu}_{\text{ptl}}$, and $\boldsymbol{\nu}_{\text{reco}}$, collectively denoted by $\boldsymbol{\nu}$ when simplifying notation. The differential cross-section in the fiducial phase space from equation (4) then becomes

$$\mathrm{d}\sigma\left(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\nu}_{\text{reco}},\boldsymbol{\nu}_{\text{ptl}},\boldsymbol{\nu}_{\text{p}}\right)=\sigma\left(\boldsymbol{\theta},\boldsymbol{\nu}_{\text{reco}},\boldsymbol{\nu}_{\text{ptl}},\boldsymbol{\nu}_{\text{p}}\right)\int\mathrm{d}\boldsymbol{z}_{\text{reco}}\int\mathrm{d}\boldsymbol{z}_{\text{ptl}}\int\mathrm{d}\boldsymbol{z}_{\text{p}}\,p\left(\boldsymbol{x},\boldsymbol{z}_{\text{reco}},\boldsymbol{z}_{\text{ptl}},\boldsymbol{z}_{\text{p}}|\boldsymbol{\theta},\boldsymbol{\nu}_{\text{reco}},\boldsymbol{\nu}_{\text{ptl}},\boldsymbol{\nu}_{\text{p}}\right)\mathrm{d}\boldsymbol{x}. \tag{12}$$

The hierarchical event representation enables a natural factorization of the pdf as

$$p\left(\boldsymbol{x},\boldsymbol{z}_{\text{reco}},\boldsymbol{z}_{\text{ptl}},\boldsymbol{z}_{\text{p}}|\boldsymbol{\theta},\boldsymbol{\nu}_{\text{reco}},\boldsymbol{\nu}_{\text{ptl}},\boldsymbol{\nu}_{\text{p}}\right)=p\left(\boldsymbol{x}|\boldsymbol{z}_{\text{reco}}\right)p\left(\boldsymbol{z}_{\text{reco}}|\boldsymbol{z}_{\text{ptl}},\boldsymbol{\nu}_{\text{reco}}\right)p\left(\boldsymbol{z}_{\text{ptl}}|\boldsymbol{z}_{\text{p}},\boldsymbol{\nu}_{\text{ptl}}\right)p\left(\boldsymbol{z}_{\text{p}}|\boldsymbol{\theta},\boldsymbol{\nu}_{\text{p}}\right). \tag{13}$$

This pdf depends on both latent and observable features [14]. In ratios, the conditional factors[1] can partially or entirely cancel, enabling efficient generation of synthetic data sets for trainng ML surrogates [17].

4.2. Semi-analytic modeling at the parton level

At the parton level, the ME generators provide a (possibly weighted) sampling of the ME-squared SMEFT terms. The generic parton-level DCR for a single process is

$$\begin{aligned}\mathrm{d}\sigma_{\text{SMEFT}}\left(\boldsymbol{z}_p|\boldsymbol{\theta},\nu_{\text{R}},\nu_{\text{F}},\boldsymbol{\nu}_{\text{PDF}}\right)\propto&\sum_{f_1,f_2,h}\left|\mathcal{M}_{\text{SMEFT}}\left(\boldsymbol{z}_p,h|\boldsymbol{\theta},\mu_{\text{R}}\left(\nu_{\text{R}}\right),\mu_{\text{F}}\left(\nu_{\text{F}}\right)\right)\right|^2\\&\times\text{PDF}\left(f_1,x_{\text{Bjorken},1},\mu_{\text{F}}\left(\nu_{\text{F}}\right),\boldsymbol{\nu}_{\text{PDF}}\right)\text{PDF}\left(f_2,x_{\text{Bjorken},2},\mu_{\text{F}}\left(\nu_{\text{F}}\right),\boldsymbol{\nu}_{\text{PDF}}\right)\mathrm{d}\boldsymbol{z}_p,\end{aligned} \tag{14}$$

where μ_{R} and μ_{F} are the renormalization and factorization scales, respectively. For single-operator insertions, the dependence on the SMEFT Wilson coefficients $\boldsymbol{\theta}$ is accurately described by a quadratic polynomial. The flavors of the incoming partons are denoted by $f_{1/2}$ and take values in $\{\mathrm{u},\bar{\mathrm{u}},\mathrm{d},\bar{\mathrm{d}},\mathrm{c},\bar{\mathrm{c}},\mathrm{s},\bar{\mathrm{s}},\mathrm{b},\bar{\mathrm{b}},\mathrm{g}\}$. Furthermore, we denote the Bjorken scaling variables by $x_{\text{Bjorken},1/2}$. The relevant latent parton-level configuration, sufficient for evaluating the parton distribution functions (PDFs) [47], is then given by $\{f_1,f_2,x_{\text{Bjorken},1},x_{\text{Bjorken},2}\}$. Equation (14) also sums over the initial and final-state helicity configurations, denoted by h. This choice removes the helicity configuration from the parton-level

[1] The factor $p(\boldsymbol{x}|\boldsymbol{z}_{\text{reco}})$ could also be conditional on nuisance parameters related to uncertainties in analysis-dependent parameters that may be involved when computing $\boldsymbol{x}$ from $\boldsymbol{z}_{\text{reco}}$. This extension is straightforward.

latent-space event representation $\boldsymbol{z}_\mathrm{p}$ and is called 'helicity-ignorant'[2] in the context of reweighted predictions [48]. A list of four-momenta then represents the final state parton-level dynamics, and we arrive at the parton-level representation

$$\boldsymbol{z}_\mathrm{p} = \left\{f_1, f_2, x_{\mathrm{Bjorken},1}, x_{\mathrm{Bjorken},2}, p_1^\mu, p_2^\mu, \ldots\right\} \tag{15}$$

if the helicity configuration is kept ('helicity-aware' [48]) and enters the particle-level simulation, h is also included. The only required change in equation (14) in that case is to use a separate differential $\mathrm{d}\boldsymbol{z}_\mathrm{p}^{(h)}$ for each helicity configuration [7].

Uncertainties in the PDFs can be expressed as variations along Hessian eigen-directions of an underlying parametrization [47], with the corresponding nuisance parameters denoted by $\boldsymbol{\nu}_\mathrm{PDF}$. These PDF variations represent different hypotheses about proton parton dynamics and have a clear physical interpretation. In contrast, the dependence on μ_R and μ_F arises from technical artifacts, namely the finite perturbative order and the separation of collinear radiation between the ME and the PDF. Despite this, we treat scale uncertainties with standard nuisance parameters ν_R and ν_F, corresponding to up and down variations around the central values $\mu_{R,0}$ and $\mu_{F,0}$. If values of ± 1 correspond to variation factors of 2, then

$$\mu_\mathrm{R}\left(\nu_\mathrm{R}\right) = 2^{\nu_\mathrm{R}}\mu_{R,0} \quad \text{and} \quad \mu_\mathrm{F}\left(\nu_\mathrm{F}\right) = 2^{\nu_\mathrm{F}}\mu_{F,0}. \tag{16}$$

We account for uncertainties in the overall normalization of the process with log-normal nuisances in section 6 and will not discuss them here. However, we include a general parton-level ad-hoc reweighting function $\alpha_\mathrm{rw}(\boldsymbol{z}_\mathrm{p}, \boldsymbol{\nu}_\mathrm{rw})$, useful when higher-order perturbative corrections significantly depend on the parton-level configuration and we want to adjust the parton-level distribution with an ad-hoc modification. The total parton-level prediction is then written as

$$\mathrm{d}\sigma\left(\boldsymbol{z}_p|\boldsymbol{\theta}, \boldsymbol{\nu}_\mathrm{p}\right) = \alpha_\mathrm{rw}\left(\boldsymbol{z}_\mathrm{p}, \boldsymbol{\nu}_\mathrm{rw}\right)\, \mathrm{d}\sigma_\mathrm{SMEFT}\left(\boldsymbol{z}_p|\boldsymbol{\theta}, \nu_\mathrm{R}, \nu_\mathrm{F}, \boldsymbol{\nu}_\mathrm{PDF}\right). \tag{17}$$

An example of this type is the modeling of the transverse top quark momenta in the $\mathrm{t\bar{t}}$ process, which is well understood from higher-order pertubation theory [49], but is not yet available from SMEFT ME generators. The nuisances $\boldsymbol{\nu}_\mathrm{rw}$ modify parameters in the reweighting function $\alpha_\mathrm{rw}(\boldsymbol{z}_p, \boldsymbol{\nu}_\mathrm{rw})$ within their uncertainties. Since such procedures are highly application-dependent, we do not elaborate further, except to note that $\alpha_\mathrm{rw}(\boldsymbol{z}_p, \boldsymbol{\nu}_\mathrm{rw})$ must always be positive. The parton-level nuisance parameters considered so far are

$$\boldsymbol{\nu}_\mathrm{p} = \left\{\nu_\mathrm{R}, \nu_\mathrm{F}, \boldsymbol{\nu}_\mathrm{PDF}, \boldsymbol{\nu}_\mathrm{rw}\right\} \tag{18}$$

and are associated with systematic effects that can be modeled semi-analytically, allowing for efficient computation.

We use equation (17) to define a parton-level pdf and the inclusive parton-level cross-section generically as

$$\mathrm{d}\sigma\left(\boldsymbol{z}_\mathrm{p}|\boldsymbol{\theta}, \boldsymbol{\nu}_\mathrm{p}\right) = \bar{\sigma}\left(\boldsymbol{\theta}, \boldsymbol{\nu}_\mathrm{p}\right)\, p\left(\boldsymbol{z}_\mathrm{p}|\boldsymbol{\theta}, \boldsymbol{\nu}_\mathrm{p}\right)\, \mathrm{d}\boldsymbol{z}_\mathrm{p}. \tag{19}$$

The bar on $\bar{\sigma}(\boldsymbol{\theta}, \boldsymbol{\nu}_\mathrm{p})$ indicates that the inclusive cross-section pertains to the entire kinematic phase-space, unaffected, e.g. by the finite detector acceptance. By definition, we have

$$\int d\sigma\left(\boldsymbol{z}_\mathrm{p}|\boldsymbol{\theta}, \boldsymbol{\nu}_\mathrm{p}\right) = \bar{\sigma}\left(\boldsymbol{\theta}, \boldsymbol{\nu}_\mathrm{p}\right) \quad \text{and} \quad \int \mathrm{d}\boldsymbol{z}_\mathrm{p}\, p\left(\boldsymbol{z}_\mathrm{p}|\boldsymbol{\theta}, \boldsymbol{\nu}_\mathrm{p}\right) = 1, \tag{20}$$

which differs from the fiducial cross-section $\sigma(\boldsymbol{\theta}, \boldsymbol{\nu})$ in equation (4) by the acceptance effects and the event selection from the subsequent modeling stages, in particular at the detector level. With an ME generator, we obtain a possibly weighted sample of identically and independently distributed events from equation (17) as

$$\left\{w_i, \boldsymbol{z}_{p,i}\right\} \overset{\text{i.i.d.}}{\sim} \bar{\sigma}\left(\boldsymbol{\theta}, \boldsymbol{\nu}_\mathrm{p}\right)\, p\left(\boldsymbol{z}_\mathrm{p}|\boldsymbol{\nu}_\mathrm{p}\right). \tag{21}$$

The overall normalization of weights w_i can be set to $\sum w_i = \bar{\sigma}(\boldsymbol{\theta}, \boldsymbol{\nu})$.

These parton-level systematic effects enable tractable simulation, allowing an inexpensive way to modify an existing sample, such as by reweighting, to approximate model parameters beyond the nominal values. Many other effects, such as the choice of different ME generators, do not allow for tractable simulation. If we model differences between such 'two point alternatives' with nuisance parameters, it becomes impossible to

[2] Helicity-aware and helicity-ignorant SMEFT predictions are compared in [7].

compute the likelihood ratio for one generator choice while sampling from another. However, in section 5, we show how to use ML to create parametrizations that interpolate between two point modeling alternatives using a nuisance parameter. This approach is crucial in practice, as uncertainties, especially those in modeling nonperturbative aspects of QCD, are often of this type.

4.3. Forward-mode event generation at particle and reconstruction level

Particle-level simulation, staged as described in section 4.1, includes the PS, initial- and final-state radiation, multiple-parton interactions, color reconnection, hadronization, the underlying event, and hadron decay. The subsequent reconstruction-level simulation models the detector interaction and event reconstruction. For each parton-level event $\{w_i, \boldsymbol{z}_{\mathrm{p},i}\}$, the particle and detector-level simulations, along with observable reconstruction, produce an event representation of the form

$$\left\{w_i, \boldsymbol{x}_i, \boldsymbol{z}_{\mathrm{reco},i}, \boldsymbol{z}_{\mathrm{ptl},i}, \boldsymbol{z}_{\mathrm{p},i}\right\}. \tag{22}$$

Due to finite detector acceptance, some events will not pass the event selection. Poor reconstruction performance near reconstruction thresholds motivates defining a fiducial region, denoted by $\mathcal{X}$. Analysis-specific selections, including reliable object-level calibrations, background reduction, and prior knowledge of SMEFT sensitivity, are incorporated into the definition of $\mathcal{X}$. We only require that for a real event, it is possible to determine if it belongs in $\mathcal{X}$. In particular, $\mathcal{X}$ can include requirements on latent variables like $\boldsymbol{z}_{\mathrm{reco}}$: online selection acceptance, thresholds on reconstructed object properties, and various data-cleaning event vetoes are part of the definition of $\mathcal{X}$, even if these variables are typically not in $\boldsymbol{x}$.

In contrast to the parton level, the particle and reconstruction-level simulation is computationally expensive, and most effects at these stages can not be simulated tractably. Event samples are, therefore, only available for a limited set of model parameters for different $\boldsymbol{\nu}_{\mathrm{ptl}}$ and $\boldsymbol{\nu}_{\mathrm{reco}}$. For each such configuration, the event sample is written as

$$\mathcal{D}(\boldsymbol{\theta},\boldsymbol{\nu}) = \left\{w_i, \boldsymbol{x}_i, \boldsymbol{z}_{\mathrm{reco},i}, \boldsymbol{z}_{\mathrm{ptl},i}, \boldsymbol{z}_{\mathrm{p},i}\right\} \overset{\text{i.i.d.}}{\sim} \sigma(\boldsymbol{\theta},\boldsymbol{\nu})\, p(\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\theta},\boldsymbol{\nu}) \quad \text{if} \quad \{\boldsymbol{x}_i, \boldsymbol{z}_{\mathrm{reco},i}\} \in \mathcal{X}, \tag{23}$$

where the total fiducial cross-section in equation (23) is

$$\sigma(\boldsymbol{\theta},\boldsymbol{\nu}) = \sum_{\boldsymbol{x}_i, \boldsymbol{z}_{\mathrm{reco},i} \in \mathcal{X}} w_i. \tag{24}$$

There is a conceptual difference between equations (21) and (23). While the parton-level distribution in equation (21) on the r.h.s. is analytically known and used in the MC sampling, equation (23) should be understood in the reverse direction. The simulated sample approximates the joint pdf in the fiducial region on the r.h.s., which is unavailable otherwise. Concretely, the formal approximation of the joint pdf is

$$\sigma(\boldsymbol{\theta},\boldsymbol{\nu})\, p(\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\theta},\boldsymbol{\nu}) \approx \sum_{\mathcal{D}(\boldsymbol{\theta},\boldsymbol{\nu})} w_i \delta(\boldsymbol{x}-\boldsymbol{x}_i)\, \delta(\boldsymbol{z}-\boldsymbol{z}_i), \tag{25}$$

and we can interpret the event weights as

$$w_i = \sigma(\boldsymbol{\theta},\boldsymbol{\nu})\, p(\boldsymbol{x}_i,\boldsymbol{z}_i|\boldsymbol{\theta},\boldsymbol{\nu}). \tag{26}$$

At NLO, the generated samples are necessarily weighted at the ME stage, and the weights can partly be negative [50]. Negative weights, in principle, invalidate the interpretation in equation (26), but equation (25) still holds, provided the large sample limit is respected.

To connect to the binned analyses, the expected yield in a given bin $\Delta\boldsymbol{x} \subset \mathcal{X}$ is given by $\lambda_{\Delta\boldsymbol{x}}(\boldsymbol{\theta},\boldsymbol{\nu}) = \mathcal{L}(\boldsymbol{\nu})\sigma_{\Delta\boldsymbol{x}}(\boldsymbol{\theta},\boldsymbol{\nu})$ with

$$\begin{aligned}\sigma_{\Delta\boldsymbol{x}}(\boldsymbol{\theta},\boldsymbol{\nu}) &= \int_{\Delta\boldsymbol{x}} \mathrm{d}\boldsymbol{x} \frac{\mathrm{d}\sigma(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\nu})}{\mathrm{d}\boldsymbol{x}} = \sigma(\boldsymbol{\theta},\boldsymbol{\nu}) \int_{\Delta\boldsymbol{x}} \mathrm{d}\boldsymbol{x}\, p(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\nu}) \\ &= \sigma(\boldsymbol{\theta},\boldsymbol{\nu}) \int_{\Delta\boldsymbol{x}} \mathrm{d}\boldsymbol{x} \int \mathrm{d}\boldsymbol{z}\, p(\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\theta},\boldsymbol{\nu}) \approx \sum_{\boldsymbol{x}_i \in \mathcal{D}(\boldsymbol{\theta},\boldsymbol{\nu}) \cap \Delta\boldsymbol{x}} w_i,\end{aligned} \tag{27}$$

where the sum extends over all events within $\mathcal{D}(\boldsymbol{\nu},\boldsymbol{\theta})$ that fall in the volume $\Delta\boldsymbol{x}$. We denote this selection by $\mathcal{D}(\boldsymbol{\theta},\boldsymbol{\nu}) \cap \Delta\boldsymbol{x}$. Cross-section weighted expectation values, used to define the loss functions in section 5, are approximated as

$$\langle \mathcal{O}(\boldsymbol{x},\boldsymbol{z}) \rangle_{\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\theta},\boldsymbol{\nu}} \equiv \int \mathrm{d}\boldsymbol{x}\, \mathrm{d}\boldsymbol{z}\, \sigma(\boldsymbol{\theta},\boldsymbol{\nu})\, p(\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\theta},\boldsymbol{\nu})\, \mathcal{O}(\boldsymbol{x},\boldsymbol{z}) \approx \sum_{\mathcal{D}(\boldsymbol{\theta},\boldsymbol{\nu})} w_i\, \mathcal{O}(\boldsymbol{x}_i,\boldsymbol{z}_i). \tag{28}$$

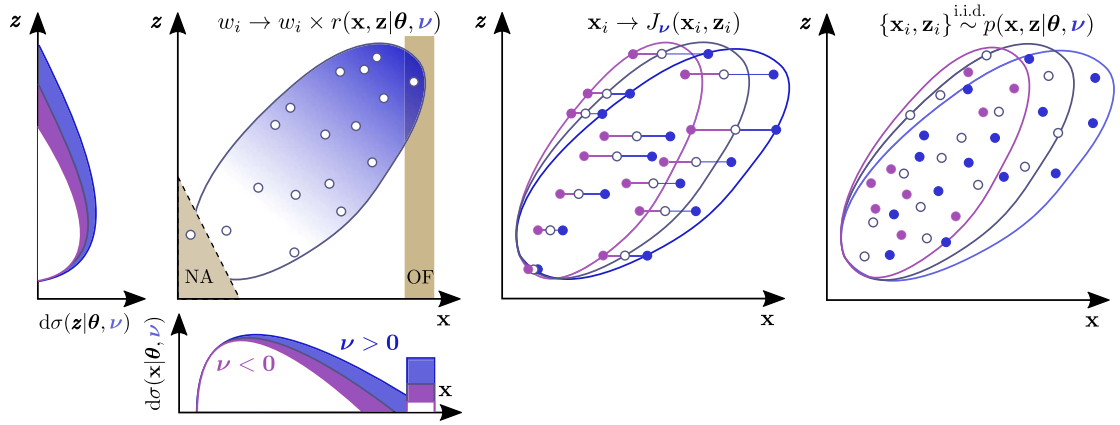

Figure 3. Illustration of the modeling of systematic effects. Left: a reweighting function, depending on $\boldsymbol{x}$ and $\boldsymbol{z}$, modifies the event weight and, thereby, tractably changes the distributions of both the latent and observed event features. Events in the region 'NA' fail acceptance, and the region 'OF' depicts an overflow bin. Middle: a function $J_{\boldsymbol{\nu}}(\boldsymbol{x}, \boldsymbol{z})$ provides variations of observed features, for example, related to uncertainties in the calibration of the underlying reconstructed objects. The white, blue, and magenta markers correspond to the nominal simulation and the up and down variations, respectively. The simulation is tractable in this case. Right: non-tractable systematic effect can only be modeled by independent sampling at different parameter points.

We emphasize that the per-event weights w_i in equation (28) are typically only known for a small set of model parameter configurations.

4.4. Synthetic data sets and tractable simulation

Simulated samples are computationally expensive. While it is always possible to obtain a simulation based on equation (23) for a specific model parameter configuration $(\boldsymbol{\theta}_0, \boldsymbol{\nu}_0)$, it is practically important to know whether we can efficiently generate a new simulated (synthetic) data set from an existing one when $\boldsymbol{\theta} \neq \boldsymbol{\theta}_0$ or $\boldsymbol{\nu} \neq \boldsymbol{\nu}_0$ for some model parameters. In such cases, the simulation is called 'tractable' for these parameters.

We discuss two tractable cases: likelihood-based reweighting and variations in the calibration of the reconstructed objects. Since SMEFT effects can also be modeled tractably, we address this separately. If a simulation is non-tractable for a model parameter, we must use equation (23) to obtain systematically varied data sets. A visualization of these approaches is shown in figure 3.

4.4.1. Uncertainties in the calibration of reconstructed objects

An important type of tractable simulation addresses uncertainties in the calibration of underlying object properties, such as jet and lepton momenta or the discriminator value of a b-tagging algorithm. Variations of $\boldsymbol{x}$ from these uncertainties are obtained by recomputing it based on modified event properties, defining a function $\boldsymbol{x}_{\boldsymbol{\nu}} = J_{\boldsymbol{\nu}}(\boldsymbol{x}, \boldsymbol{z})$. This function provides adjusted values of $\boldsymbol{x}$ that depend on latent object-level and event properties, so $J_{\boldsymbol{\nu}}$ also depends on $\boldsymbol{z}$.

Evaluating $J_{\boldsymbol{\nu}}$ provides information on how the observation changes as a function of the model parameters, but it does not give the likelihood or cross-section ratio as a function of the model parameters for a fixed observation, which is needed for equation (11). For simulated data at a reference parameter point $\{w_i, \boldsymbol{x}_i, \boldsymbol{z}_i\} \overset{\text{iid}}{\sim} p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}_0, \boldsymbol{\nu}_0)$, applying $J_{\boldsymbol{\nu}}$ instead implies

$$p(\boldsymbol{x}_i, \boldsymbol{z}_i|\boldsymbol{\theta}_0, \boldsymbol{\nu}_0) = p\left(J_{\boldsymbol{\nu}}(\boldsymbol{x}_i, \boldsymbol{z}_i), \boldsymbol{z}_i|\boldsymbol{\theta}_0, \boldsymbol{\nu}\right), \tag{29}$$

meaning the joint likelihood remains unchanged with $\boldsymbol{\nu}$ when we simultaneously modify the observation to $\boldsymbol{x}_{\boldsymbol{\nu},i} = J_{\boldsymbol{\nu}}(\boldsymbol{x}_i, \boldsymbol{z}_i)$. Synthetic data samples can be efficiently generated as

$$\mathcal{D}(\boldsymbol{\theta}_0, \boldsymbol{\nu}) = \left\{w_i, \boldsymbol{x}_{\boldsymbol{\nu},i} = J_{\boldsymbol{\nu}}(\boldsymbol{x}_i, \boldsymbol{z}_i), \boldsymbol{z}_i\right\}_{i=1}^{N_{\text{sim}}} \quad \text{for all} \quad \{w_i, \boldsymbol{x}_i, \boldsymbol{z}_i\} \in \mathcal{D}(\boldsymbol{\theta}_0, \boldsymbol{\nu}_0), \tag{30}$$

and are enough to learn a DCR surrogate as a function of $\boldsymbol{\nu}$, as discussed in section 5. We assume that calibration-type uncertainties are independent of the POIs, so $\boldsymbol{\theta}_0$ appears on both sides of equation (29).

4.4.2. Synthetic data from event-reweighting

When the change in the likelihood of observing an event as a function of a specific model parameter can be computed without resampling the pdf, we can generate reweighted synthetic data sets as

$$\mathcal{D}(\boldsymbol{\theta}, \boldsymbol{\nu}) = \left\{w_i \times r(\boldsymbol{x}_i, \boldsymbol{z}_i|\boldsymbol{\theta}, \boldsymbol{\nu}, \boldsymbol{\theta}_0, \boldsymbol{\nu}_0), \boldsymbol{x}_i, \boldsymbol{z}_i\right\}_{i=1}^{N_{\text{sim}}} \quad \text{for all} \quad \{w_i, \boldsymbol{x}_i, \boldsymbol{z}_i\} \in \mathcal{D}(\boldsymbol{\theta}_0, \boldsymbol{\nu}_0). \tag{31}$$

A reweighting function $r(\boldsymbol{x}_i, \boldsymbol{z}_i|\boldsymbol{\theta}, \boldsymbol{\nu}, \boldsymbol{\theta}_0, \boldsymbol{\nu}_0)$ must be available for the corresponding model parameter and can depend on both observables and latent features. Important tractable cases occur at the parton level, where access to the analytic form of ME-squared terms allows computation of the joint likelihood ratio. In addition to parton-level parametrizations of the form

$$r(\boldsymbol{x}_i, \boldsymbol{z}_i|\boldsymbol{\theta}_0, \boldsymbol{\nu}) = \alpha_{\text{rw}}\left(\boldsymbol{z}_{i,\text{p}}, \boldsymbol{\nu}_{\text{rw}}\right) \tag{32}$$

designed to approximate higher-order perturbative corrections as in equation (17), it is useful to describe uncertainties related to other theoretical inaccuracies with nuisance parameters. For example, the predicted rates of events with high particle-level jet multiplicity ($N_{\text{gen-jet}}$) depend on the specifics of matching between the ME calculation and the PS [51], often with significant uncertainties. To address this, we can introduce ad-hoc uncertainties for events with different particle-level jet multiplicities through the variation

$$r(\boldsymbol{x}_i, \boldsymbol{z}_i|\boldsymbol{\theta}_0, \boldsymbol{\nu}) = 1 + \alpha_1^{\nu_{\text{gen-jet}}}\delta_{1,N_{\text{gen-jet}}} + \alpha_2^{\nu_{\text{gen-jet}}}\delta_{2,N_{\text{gen-jet}}} + \cdots \tag{33}$$

where α_1, α_2, etc are constants, and $\nu_{\text{gen-jet}}$ is the associated nuisance parameter.

A special case is reweighting functions that depend solely on $\boldsymbol{x}$. In this case, we can access the DCR as a function of the observed features without requiring any learning. For example, consider a selection with a fixed lepton multiplicity N_ℓ, where the reconstruction efficiency has a relative uncertainty $\Delta\text{SF}(\ell)/\text{SF}(\ell)$, and a corrective scale factor $\text{SF}(\ell)$ is already included in the nominal simulation. We treat the uncertainty $\Delta\text{SF}(\ell)$ with a nuisance parameter ν_ℓ. If $\boldsymbol{x}$ includes the properties of the leptons, allowing $\Delta\text{SF}(\ell)$ to be computed solely from $\boldsymbol{x}$, we have

$$r(\boldsymbol{x}_i|\nu_\ell) = \prod_{i=1}^{N_\ell}\left(1 + \frac{\Delta\text{SF}\left(\ell_i\right)}{\text{SF}\left(\ell_i\right)}\right)^{\nu_\ell}, \tag{34}$$

which provides the DCR as a function of ν_ℓ without needing a surrogate. However, if $\boldsymbol{x}$ alone is insufficient to compute the scale factors, the right-hand side of equation (34) represents $r(\boldsymbol{x}_i, \boldsymbol{z}_i|\nu)$ rather than $r(\boldsymbol{x}_i|\nu)$, and a surrogate is required.

4.4.3. SMEFT modeling

SMEFT effects can be modeled with synthetic data, which initially motivated the development of optimal SMEFT observables [13–18]. The ME in equation (14) can be efficiently recomputed as a function of the POIs $\boldsymbol{\theta}$. The SMEFT dependence of the DCR is

$$\begin{aligned} &r(\boldsymbol{x}_i, \boldsymbol{z}_i|\boldsymbol{\theta}, \boldsymbol{\theta}_0) \qquad (35)\\ &= \frac{\sigma(\boldsymbol{\theta})}{\sigma(\text{SM})}\frac{p\left(\boldsymbol{x}_i, \boldsymbol{z}_{\text{reco},i}, \boldsymbol{z}_{\text{ptl},i}, \boldsymbol{z}_{\text{p},i}|\boldsymbol{\theta}\right)}{p\left(\boldsymbol{x}_i, \boldsymbol{z}_{\text{reco},i}, \boldsymbol{z}_{\text{ptl},i}, \boldsymbol{z}_{\text{p},i}|\text{SM}\right)} = \frac{\sigma(\boldsymbol{\theta})}{\sigma(\text{SM})}\frac{p\left(\boldsymbol{x}|\boldsymbol{z}_{\text{reco}}\right)}{p\left(\boldsymbol{x}|\boldsymbol{z}_{\text{reco}}\right)}\frac{p\left(\boldsymbol{z}_{\text{reco}}|\boldsymbol{z}_{\text{ptl}}\right)}{p\left(\boldsymbol{z}_{\text{reco}}|\boldsymbol{z}_{\text{ptl}}\right)}\frac{p\left(\boldsymbol{z}_{\text{ptl}}|\boldsymbol{z}_{\text{p}}\right)}{p\left(\boldsymbol{z}_{\text{ptl}}|\boldsymbol{z}_{\text{p}}\right)}\frac{p\left(\boldsymbol{z}_{\text{p}}|\boldsymbol{\theta}\right)}{p\left(\boldsymbol{z}_{\text{p}}|\text{SM}\right)}\\ &= \frac{\sigma(\boldsymbol{\theta})}{\sigma(\text{SM})}\frac{p\left(\boldsymbol{z}_{\text{p},i}|\boldsymbol{\theta}\right)}{p\left(\boldsymbol{z}_{\text{p},i}|\text{SM}\right)} = \frac{\left|\mathcal{M}\left(\boldsymbol{z}_{\text{p},i}, \boldsymbol{\theta}\right)\right|^2}{\left|\mathcal{M}\left(\boldsymbol{z}_{\text{p},i}, \text{SM}\right)\right|^2} = 1 + \theta_m r^{(m)}\left(\boldsymbol{z}_{\text{p},i}\right) + \theta_m\theta_n r^{(mn)}\left(\boldsymbol{z}_{\text{p},i}\right), \qquad (36)\end{aligned}$$

where we have omitted the nuisance parameter dependence, as the numerator and denominator are evaluated for the same $\boldsymbol{\nu}_0$. The conditional probabilities in the third term, which are not tractable, cancel in the ratio with excellent accuracy. The remainder is the parton-level DCR, which is available at the level of the ME generator. Thus, $r(\boldsymbol{x}_i, \boldsymbol{z}_i|\boldsymbol{\theta}, \boldsymbol{\theta}_0)$ does not depend on $\boldsymbol{x}_i$. By calculating the per-event polynomial coefficients $r^{(m)}(\boldsymbol{z}_{\text{p},i})$ and $r^{(mn)}(\boldsymbol{z}_{\text{p},i})$ with $m, n = 1, \ldots, N_{\boldsymbol{\theta}}$ from the ME-squared terms at various $\boldsymbol{\theta}$ values, we can construct a parametrization of $r(\boldsymbol{x}_i, \boldsymbol{z}_i|\boldsymbol{\theta}, \boldsymbol{\theta}_0)$ valid across the entire SMEFT parameter space, making synthetic data sets $\mathcal{D}(\boldsymbol{\theta}, \boldsymbol{\nu}_0)$ readily available. We provide further details in appendix A. Note that the SM point in the denominator is not unique; equation (36) can be applied for any $\boldsymbol{\theta}_0$, allowing for simulation at EFT parameter points other than the SM.

4.5. Large sample limit and overflow bins

Learning surrogates of the likelihood ratio requires sufficient simulated data, and in the following sections, we assume the large sample limit when minimizing loss functions. For any finite simulated data set, observables with energy units often imply a threshold beyond which simulation becomes too sparse. Since SMEFT effects can increase with energy, removing events in the tails of energetic variable distributions is generally counterproductive. Instead, for each such variable in $\boldsymbol{x}$, we can define a threshold beyond which we do not fully trust the modeling of the differential cross-section but still find acceptable uncertainties in the cumulative yield.

We accumulate all events with features above certain thresholds in several 'overflow bins' (OF) and treat these bins using standard Poisson likelihoods. Suppose x_n is a feature in the vector of observables $\boldsymbol{x}'$ representing the event before introducing overflow. In that case, we must address the fact that synthetic data insufficiently samples $p(\boldsymbol{x}'|x_n > x_{n,0}, \boldsymbol{\theta}, \boldsymbol{\nu})$. The simplest solution is to drop all event features for $x_n > x_{n,0}$ and represent the event only by its presence in the overflow of x_n, i.e.

$$\boldsymbol{x}' = \begin{cases} \boldsymbol{x} & x_n < x_{n,0} \\ \mathrm{OF}_n & \text{otherwise.} \end{cases} \tag{37}$$

For a number of overflow bins (N_{OF}), one for each required observable, we reduce the observation to the counting variable $N(\mathrm{OF}_n)$ instead of using fully unbinned information. The number N_{OF} characterizes the measurement, while the observed number of events in the n-th overflow bin $N(\mathrm{OF}_n)$ characterizes the observation. Explicitly, $\mathcal{D}' = \mathcal{D} \cup \{N(\mathrm{OF}_n)\}_{n=1}^{N_{\mathrm{OF}}}$. The likelihood for the overflow bin observation follows from equation (5) as a Poisson factor for each overflow bin as

$$L(\mathcal{D}'|\boldsymbol{\theta}, \boldsymbol{\nu}) = L(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\nu}) \times \prod_{n=1}^{N_{\mathrm{OF}}} \mathrm{P}\left(N(\mathrm{OF}_n)\,|\lambda(\mathrm{OF}_n|\boldsymbol{\theta}, \boldsymbol{\nu})\right). \tag{38}$$

As assumed, the predicted yield in each overflow bin as a function of the POIs and nuisances, denoted by $\lambda(\mathrm{OF}_n|\boldsymbol{\theta}, \boldsymbol{\nu})$, is available from simulation with sufficient precision. This additional factor is a standard binned likelihood term and should be handled as in traditional binned SMEFT analyses. To simplify the formulas in the following sections, we assume that this factor is included in equation (5) and that $\boldsymbol{x} \in \mathcal{X}$ implies the event is not in any overflow bin.

5. Learning from simulation

With procedures in place for obtaining simulated and synthetic data sets, we now outline how parametrizations can be learned using common loss functions, such as cross-entropy loss. We consider a general expressive function $\hat{f}(\boldsymbol{x})$, without specifying its implementation; it could be a neural network, the BPT from section 5.4, or any other trainable multivariate predictive function.

5.1. Likelihood-ratio trick and cross-entropy loss

We begin with the well-known 'likelihood-ratio trick,' which underpins the learning tasks in this work: a sufficiently expressive machine trained on a classification task learns a (monotonic function of) the likelihood ratio. If the training data is normalized by the differential cross-section, the classifier learns the DCR. This fact is at the heart of learning techniques for parametric surrogates. Let us take two fixed hypotheses $(\boldsymbol{\theta}_0, \boldsymbol{\nu}_0)$ and $(\boldsymbol{\theta}_1, \boldsymbol{\nu}_1)$, obtained either by independent simulation or produced synthetically as described in section 4.4, and minimize the cross-entropy loss function

$$L_{\mathrm{CE}}\left[\hat{f}\right] = -\langle \log \hat{f}(\boldsymbol{x}) \rangle_{\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\theta}_0,\boldsymbol{\nu}_0} - \langle \log\left(1 - \hat{f}(\boldsymbol{x})\right) \rangle_{\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\theta}_1,\boldsymbol{\nu}_1}. \tag{39}$$

The minimum of equation (39) for $\hat{f}(\boldsymbol{x})$ is attained where the function derivative vanishes,

$$\frac{\delta L_{\mathrm{CE}}\left[\hat{f}\right]}{\delta \hat{f}(\boldsymbol{x})} = 0. \tag{40}$$

Because $\hat{f}(\boldsymbol{x})$ does not depend on $\boldsymbol{z}$ by construction, we can formally integrate over the latent configuration using $\int \mathrm{d}\boldsymbol{z} p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}, \boldsymbol{\nu}) = p(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{\nu})$, separately for the summands in equation (39). The solution is then expressed in terms of latent-space integrals as

$$f^*_{\mathrm{CE}}(\boldsymbol{x}) \equiv \mathrm{argmin}_{\hat{f}} L_{\mathrm{CE}}\left[\hat{f}\right] = \left(1 + \frac{\sigma(\boldsymbol{\theta}_1, \boldsymbol{\nu}_1) \int \mathrm{d}\boldsymbol{z} p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}_1, \boldsymbol{\nu}_1)}{\sigma(\boldsymbol{\theta}_0, \boldsymbol{\nu}_0) \int \mathrm{d}\boldsymbol{z} p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}_0, \boldsymbol{\nu}_0)}\right)^{-1} = \left(1 + \frac{\mathrm{d}\sigma(\boldsymbol{x}|\boldsymbol{\theta}_1, \boldsymbol{\nu}_1)}{\mathrm{d}\sigma(\boldsymbol{x}|\boldsymbol{\theta}_0, \boldsymbol{\nu}_0)}\right)^{-1}. \tag{41}$$

This expression is a monotonous function of the DCR for two fixed choices of the model parameters and can be rearranged to

$$\frac{\mathrm{d}\sigma(\boldsymbol{x}|\boldsymbol{\theta}_1, \boldsymbol{\nu}_1)}{\mathrm{d}\sigma(\boldsymbol{x}|\boldsymbol{\theta}_0, \boldsymbol{\nu}_0)} = \frac{1}{f^*_{\mathrm{CE}}(\boldsymbol{x})} - 1. \tag{42}$$

Alternative loss functions and their minima are discussed in appendix B. The simulation-based approximation of equation (39), suitable for a concrete implementation in computer code, is

$$L_{\mathrm{CE}}\left[\hat{f}\right] \approx -\sum_{\mathcal{D}(\boldsymbol{\theta}_0,\boldsymbol{\nu}_0)} w_i \log \hat{f}(\boldsymbol{x}_i) - \sum_{\mathcal{D}(\boldsymbol{\theta}_1,\boldsymbol{\nu}_1)} w_i \log\left(1-\hat{f}(\boldsymbol{x}_i)\right), \tag{43}$$

and explicitly uses two different samples in the two terms. For the reweighting-based tractable effects in sections 4.4.2 and 4.4.3, the per-event joint likelihood is available, and we can use equation (31) to rewrite the cross-entropy loss with a single pdf as

$$L_{\mathrm{CE}}\left[\hat{f}\right] = -\int \mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{z}\,\sigma(\boldsymbol{\theta}_0,\boldsymbol{\nu}_0)\,p(\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\theta}_0,\boldsymbol{\nu}_0)\left(\log\hat{f}(\boldsymbol{x}) + r(\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\theta}_1,\boldsymbol{\nu}_1,\boldsymbol{\theta}_0,\boldsymbol{\nu}_0)\log\left(1-\hat{f}(\boldsymbol{x})\right)\right). \tag{44}$$

The two terms in equation (44) separately agree with the two expectations in equation (39). The approximation of equation (44) for a synthetic data set is

$$L_{\mathrm{CE}}\left[\hat{f}\right] \approx -\sum_{\mathcal{D}(\boldsymbol{\theta}_0,\boldsymbol{\nu}_0)} w_i \left(\log\hat{f}(\boldsymbol{x}_i) + r(\boldsymbol{x}_i,\boldsymbol{z}_i|\boldsymbol{\theta}_1,\boldsymbol{\nu}_1,\boldsymbol{\theta}_0,\boldsymbol{\nu}_0)\log\left(1-\hat{f}(\boldsymbol{x}_i)\right)\right). \tag{45}$$

The main difference to equation (43) is the absence of independent stochastic fluctuations in the two terms. In both cases, $\hat{f}(\boldsymbol{x})$ is implemented as a finitely but sufficiently expressive ML algorithm, approximating the exact solution. We denote this approximation by

$$\hat{f}(\boldsymbol{x}) \simeq \left(1+\frac{\mathrm{d}\sigma(\boldsymbol{x}|\boldsymbol{\theta}_1,\boldsymbol{\nu}_1)}{\mathrm{d}\sigma(\boldsymbol{x}|\boldsymbol{\theta}_0,\boldsymbol{\nu}_0)}\right)^{-1}. \tag{46}$$

5.2. Machine-learning systematic parametrizations

To learn parametrizations, we replace $\hat{f}$ with a suitable parametric ansatz that captures the $\boldsymbol{\nu}$ dependence. The loss function is summed over a set of model parameter points (base points), denoted by $\mathcal{V}$. The parametrization can be determined using synthetic data from a sufficient number of base points. We omit $\boldsymbol{\theta}$ in the formulas, as we will factorize this dependence in section 6. The fully calibrated SM parameter point serves as the reference, $\boldsymbol{\nu}_0 = \mathbf{0}$. The ansatz

$$\hat{f}(\boldsymbol{x}) = \frac{1}{1+\exp\left(\hat{T}(\boldsymbol{x}|\boldsymbol{\nu})\right)} \tag{47}$$

eliminates the monotonous dependence from equation (41). The ML estimate of the DCR is then $\hat{S}(\boldsymbol{x}|\boldsymbol{\nu}) = \exp(\hat{T}(\boldsymbol{x}|\boldsymbol{\nu}))$. The exponential function removes the necessity of ensuring that $\hat{S}(\boldsymbol{x}|\boldsymbol{\nu})$ must be positive. Inserting equation (47) into equation (39), leads to

$$L_{\mathrm{CE}}\left[\hat{T}(\boldsymbol{x}|\boldsymbol{\nu})\right] = \left\langle \mathrm{Soft}^+\left(\hat{T}(\boldsymbol{x}|\boldsymbol{\nu})\right)\right\rangle_{\boldsymbol{x},\boldsymbol{z}|\boldsymbol{0}} + \left\langle \mathrm{Soft}^+\left(-\hat{T}(\boldsymbol{x}|\boldsymbol{\nu})\right)\right\rangle_{\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\nu}} \tag{48}$$

where $\mathrm{Soft}^+(x) = \log(1+\exp(x))$. Next, we approximate the logarithm of the DCR with a polynomial ansatz in $\boldsymbol{\nu}$ in terms of coefficient functions as

$$\hat{T}(\boldsymbol{x}|\boldsymbol{\nu}) = \nu_a \hat{\Delta}_a(\boldsymbol{x}) + \nu_a\nu_b\hat{\Delta}_{a,b}(\boldsymbol{x}) + \ldots. \tag{49}$$

The ellipsis indicates that cubic or higher terms can be added as needed, allowing to parametrize the systematic effects, in principle, with arbitrary precision. The functional form is equivalent to the ansatz in [17, 22]. Notably, we include the possibility that some of the coefficient functions are chosen to be absent.

To determine $\hat{\Delta}_a(\boldsymbol{x})$, $\hat{\Delta}_{ab}(\boldsymbol{x})$, etc via a suitable loss function, we note that equation (49) is a *linear* equation; only the coefficients in this system are polynomial in $\boldsymbol{\nu}$. Without loss of generality, we assume triangular coefficient functions $\hat{\Delta}_{abc\ldots}(\boldsymbol{x})$, i.e. $\hat{\Delta}_{abc}(\boldsymbol{x}) = 0$ unless $a \leqslant b \leqslant c$, etc and we denote their total number by N_Δ. To reduce the notational clutter, we next introduce a multi-index[3] $A = 1,\ldots,N_\Delta$. In the most general case, A labels the set $\{a,(ab),(abc),\ldots\}$ where a labels the $N_{\boldsymbol{\nu}}$ linear terms, (ab) the $N_{\boldsymbol{\nu}}(N_{\boldsymbol{\nu}}+1)/2$ quadratic terms, etc. For a given nuisance-parameter point $\boldsymbol{\nu}$, we similarly write $\nu_A = \{\nu_a, \nu_a\nu_b, \nu_a\nu_b\nu_c, \ldots\}_A$, where each element corresponds to one of the coefficient functions. This notation simplifies equation (49) to

$$\hat{T}(\boldsymbol{x}|\boldsymbol{\nu}) = \nu_A \hat{\Delta}_A(\boldsymbol{x}). \tag{50}$$

[3] We use the Einstein sum convention for the nuisance parameter index, labeled by a, b, ..., as well as for the multi-index labeled by A, B, ...

With a sufficiently large number of base points and the corresponding, possibly synthetic, data sets, we add up copies of the loss function in equation (48), one for each $\boldsymbol{\nu} \in \mathcal{V}$,

$$L\left[\hat{\Delta}_A(\boldsymbol{x})\right] = \sum_{\boldsymbol{\nu}\in\mathcal{V}} L_{\text{CE}}\left[\nu_A \hat{\Delta}_A(\boldsymbol{x})\right] \tag{51}$$

$$= \sum_{\boldsymbol{\nu}\in\mathcal{V}} \left(\left\langle \text{Soft}^+\left(\nu_A \hat{\Delta}_A(\boldsymbol{x})\right)\right\rangle_{\boldsymbol{x},\boldsymbol{z}|\boldsymbol{0}} + \left\langle \text{Soft}^+\left(-\nu_A \hat{\Delta}_A(\boldsymbol{x})\right)\right\rangle_{\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\nu}} \right) \tag{52}$$

It is straightforward to show that the minimum approximates the DCR as

$$\hat{S}(\boldsymbol{x}|\boldsymbol{\nu}) = \exp\left(\nu_A \hat{\Delta}_A(\boldsymbol{x})\right) \simeq \frac{\mathrm{d}\sigma(\boldsymbol{x}|\boldsymbol{\nu})}{\mathrm{d}\sigma(\boldsymbol{x}|\boldsymbol{0})}, \tag{53}$$

if the base points $\mathcal{V}$ span the space of the nuisance parameters, i.e. the $N_{\boldsymbol{\nu}} \times N_\Delta$-dimensional matrix of the base-point coordinates $\{\nu_A\}_{\boldsymbol{\nu}\in\mathcal{V}}$ has at least rank N_Δ. This implies, in particular, that we need at least as many base points and synthetic training data sets as there are independent coefficient functions.

Each term in the sum in equation (52) contains two expectations; note that the first expectation in each term corresponds to the SM. The coefficient functions $\hat{\Delta}_A(\boldsymbol{x})$ can be implemented as neural networks or any other trainable regressor. In most cases, the factorization of systematic effects is a reliable simplification, so the number of coefficient functions that need to be learned simultaneously remains small. For a single nuisance, one or two coefficient functions are enough to achieve linear or quadratic accuracy in equation (53), which is usually sufficient. Higher-order terms can be added as needed.

5.3. Two-point alternatives

For some systematic effects, like binary generator choices, no tractable simulation exists–only two alternate simulations. In this case, the sum over $\mathcal{V}$ in equation (52) becomes trivial, simplifying the learning task. A single nuisance parameter $\nu_{2\text{P}}$ interpolates between the nominal choice ($\nu_{2\text{P}} = 0$) and the alternative ($\nu_{2\text{P}} = 1$). Without predictions for more values, we can only learn a linear approximation. With $\nu_{2\text{P}} = 1$ as the sole value in $\mathcal{V}$, we get

$$L\left[\hat{\Delta}\right] = \langle \text{Soft}^+\left(\hat{\Delta}(\boldsymbol{x})\right)\rangle_{\boldsymbol{x},\boldsymbol{z}|0} + \left\langle \text{Soft}^+\left(-\hat{\Delta}(\boldsymbol{x})\right)\right\rangle_{\boldsymbol{x},\boldsymbol{z}|1} \tag{54}$$

where $\hat{\Delta}(\boldsymbol{x})$ is a single-valued coefficient function. Minimization provides an estimate

$$\hat{S}_{2\text{P}}(\boldsymbol{x}|\nu_{2\text{P}}) = \exp\left(\nu_{2\text{P}} \hat{\Delta}(\boldsymbol{x})\right) \simeq \left(\frac{\mathrm{d}\sigma_1(\boldsymbol{x})}{\mathrm{d}\sigma_0(\boldsymbol{x})}\right)^{\nu_{2\text{P}}}, \tag{55}$$

which is an $\boldsymbol{x}$-dependent linear interpolation of the logarithm of the DCR.

When profiling the nuisance parameter $\nu_{2\text{P}}$, it takes values other than 0 and 1, even though predictions are only well-defined at these points. Is the interpolation meaningful during profiling? This is a modeling question and cannot be resolved by statistical or ML methodology. As in the binned case, we generally recommend avoiding two-point alternatives and instead using a single model with meaningful and flexible parameters. Two-point alternatives, such as using alternative generators, are beneficial as cross-checks. When profiling the effects of $\hat{S}_{2\text{P}}(\boldsymbol{x}|\nu_{2\text{P}})$, it should be ensured that its impact is not substantial or dominant; otherwise, the validity of the measurement could be in doubt. Regardless of the modeling decision, two-point alternatives are accounted for by equation (55).

5.4. The BPT algorithm

The coefficient functions in equation (50) could be implemented using standard neural networks. However, with hundreds of nuisance parameters in a realistic analysis, it is advantageous to develop a low-maintenance, flexible algorithm to separately learn the numerous systematic dependencies. In the following, we describe a tree-boosting regressor, the BPT, designed for learning systematic effects. Its complete derivation is provided in appendix C.

Boosted tree algorithms have a strong track record in classification and regression tasks and were recently applied to novel searches for resonant phenomena [52]. They use an additive sequence of weak learners, each generating a coarsely binned prediction based on hierarchical phase-space partitioning, which is computationally efficient. Each tree's terminal nodes are linked to a predictive function that varies non-linearly across the phase-space boundaries of these nodes. Here, we extend standard tree-based regression algorithms, such as those in TMVA [53], by introducing a more flexible terminal-node predictor

that provides the DCR to arbitrary order in the expansion in $\boldsymbol{\nu}$. The summed prediction from these weak learners, trained iteratively through boosting, is both smooth and arbitrarily expressive.

The simplicity of the weak learner leads to relatively mild failure modes. For instance, if a tree is trained with insufficient data, it cannot extrapolate incorrectly to phase-space regions beyond the training data; it will simply predict the value of the highest populated bin it finds, respecting regularization requirements, such as a minimum number of events per terminal node.

Additionally, a tree algorithm with axis-linear node splits, like the 'Classification and Regression Tree' (CART) [54] algorithm, does not interpolate stepwise features in the training data. If features take only discrete values, the algorithm will partition phase space based on selections at these discrete values. Therefore, features related to object multiplicity need no special handling. Replacing nominal training data with digitized values according to a chosen binning can serve as validation, allowing sensitivity comparisons between unbinned and binned reference results.

The trees' terminal nodes can be associated with more complex quantities than simple class probabilities or regression values, as shown in applications learning polynomial SMEFT dependence [9, 10]. We exploit this flexibility for developing a boosted tree algorithm where terminal nodes of the weak learner are linked to parametrizations of systematic effects of the training data in the nodes. With the tools from section 5.2, the BPT provides a tree-based estimate $\hat{T}(\boldsymbol{x}|\boldsymbol{\nu})$ of the logarithm of the DCR in the polynomial expansion

$$\hat{T}(\boldsymbol{x}|\boldsymbol{\nu}) = \nu_A \hat{\Delta}_A(\boldsymbol{x}) \simeq \log \frac{\mathrm{d}\sigma(\boldsymbol{x}|\boldsymbol{\nu})}{\mathrm{d}\sigma(\boldsymbol{x}|\mathrm{SM})}. \tag{56}$$

So that $\hat{S}(\boldsymbol{x}|\boldsymbol{\nu}) = \exp(\hat{T}(\boldsymbol{x}|\boldsymbol{\nu}))$. The free parameters in $\hat{\Delta}_A(\boldsymbol{x})$ are trained with the CE loss function in equation (52) by an iterative boosting algorithm. It fits the weak learners of an additive expansion, one at a time, to the pseudo-residuals of the preceding boosting iteration. The complete construction is discussed in appendix C. Here, we describe the resulting algorithm.

During training, each weak learner captures only part of the total parameter dependence in each terminal node, with this fraction controlled by the algorithm's learning rate. We set a number B of boosting iterations and choose learning rates $0 < \eta^{(b)} < 1$ for $b = 1, \ldots, B$ to form an additive expansion of $\hat{T}(\boldsymbol{x})$ in terms of the weak learners. The $\eta^{(b)}$ can be chosen constant, and values between 10^{-3} and $3 \cdot 10^{-1}$ for this universal learning rate have proven efficient. At each iteration b, the weak learner is a tree with terminal nodes corresponding to phase-space partitioning-a set of non-overlapping regions $\mathcal{J}^{(b)}$ that together cover $\mathcal{X}$. This leads to

$$\hat{\Delta}_A(\boldsymbol{x}) = \sum_{b=1}^{B} \eta^{(b)} \sum_{j \in \mathcal{J}^{(b)}} \mathbb{1}_j(\boldsymbol{x}) \hat{\Delta}_{A,j}^{(b)} \tag{57}$$

where the indicator function $\mathbb{1}_j(\boldsymbol{x})$ equals one if $\boldsymbol{x}$ is in the phase-space region of terminal node j and zero otherwise. Training iteration b involves finding the partitioning $\mathcal{J}^{(b)}$ whose terminal-node predictions minimize the loss. Each terminal node prediction is based on constants $\hat{\Delta}_{A,j}^{(b)}$, which are best-fit polynomial coefficients approximating the nuisance parameter dependence in terminal node j. These coefficients, labeled by the multi-index A, are determined from the training sets $\mathcal{D}_{\mathbf{0}}$ and $\mathcal{D}_{\boldsymbol{\nu}}^{(b)}$, where we have one $\mathcal{D}_{\boldsymbol{\nu}}^{(b)}$ for each $\boldsymbol{\nu} \in \mathcal{V}$. We initialize with $\hat{\Delta}_{A,j}^{(0)} = 0$.

To proceed from iteration $b-1$ to b, we remove a fraction $\eta^{(b)}$ of the previous iteration's fit result from the training data. Since we estimate the logarithm of the DCR, the reweighting

$$\mathcal{D}_{\boldsymbol{\nu}}^{(b)} = \left\{ \exp\left(-\eta^{(b-1)} t^{(b-1)*}(\boldsymbol{x}_i|\boldsymbol{\nu})\right) w_i^{(b-1)}, \boldsymbol{x}_i, \boldsymbol{z}_i \right\} \text{ for all } \left\{ w^{(b-1)}, \boldsymbol{x}_i, \boldsymbol{z}_i \right\} \in \mathcal{D}_{\boldsymbol{\nu}}^{(b-1)} \text{ for all } \boldsymbol{\nu} \in \mathcal{V} \tag{58}$$

Produces the corresponding $\mathcal{D}_{\boldsymbol{\nu}}^{(b)}$. The nominal SM training sample $\mathcal{D}_{\mathbf{0}}$ is unchanged.

The quantity in the exponent is the best fit at iteration $b-1$,

$$t^{(b-1)*}(\boldsymbol{x}|\boldsymbol{\nu}) = \nu_A \sum_{j \in \mathcal{J}^{(b-1)}} \mathbb{1}_j(\boldsymbol{x}) \hat{\Delta}_{A,j}^{(b-1)}. \tag{59}$$

Whose polynomial coefficients also appear on the r.h.s. of equation (57). To obtain $\hat{\Delta}_{A,j}^{(b)}$, we use the new training data to predict per-node cross-section values

$$\sigma_{j,0}^{(b)} = \sum_{(\boldsymbol{x}_i, w_i) \in \mathcal{D}_{\mathbf{0}}^{(b)} \cap \Delta \boldsymbol{x}_j} w_i \qquad \text{and} \qquad \sigma_{j,\boldsymbol{\nu}}^{(b)} = \sum_{(\boldsymbol{x}_i, w_i) \in \mathcal{D}_{\boldsymbol{\nu}}^{(b)} \cap \Delta \boldsymbol{x}_j} w_i. \tag{60}$$

Algorithm 1. Boosted Parametric Tree (BPT) for learning of systematic uncertainties.

Require: base points $\boldsymbol{\nu} \in \mathcal{V}$, sample $\mathcal{D}_\mathbf{0}$ and $\mathcal{D}_{\boldsymbol{\nu}}$ for all $\boldsymbol{\nu} \in \mathcal{V}$,
boosting iterations B, learning rates $0 \leqslant \eta^{(b)} \leqslant 1$ for $b = 1, \ldots, B$.
Ensure: $\sum_{\boldsymbol{\nu}\in\mathcal{V}} \nu_A \nu_B$ has full rank
$t_{\boldsymbol{\nu}}^{(0)*}(\boldsymbol{x}) \leftarrow 0$
$\hat{T}_{\boldsymbol{\nu}}^{(0)}(\boldsymbol{x}) \leftarrow 0$
$\mathcal{D}_{\boldsymbol{\nu}}^{(0)} \leftarrow \mathcal{D}_{\boldsymbol{\nu}}$ for all $\boldsymbol{\nu} \in \mathcal{V}$
for $b = 1, \ldots, B$ **do**
  $\mathcal{D}_{\boldsymbol{\nu}}^{(b)} \leftarrow \left\{ w_i^{(b)} \leftarrow \exp(-\eta^{(b-1)} t_{\boldsymbol{\nu}}^{(b-1)*}(\mathbf{x_i}))\, w_i^{(b-1)}, \boldsymbol{x}_i, \boldsymbol{z}_i \right\}$ for all $\{w^{(b-1)}, \boldsymbol{x}_i, \boldsymbol{z}_i\} \in \mathcal{D}_{\boldsymbol{\nu}}^{(b-1)}$
  $\mathcal{J}^{(b)} \leftarrow \text{argmin}_{\mathcal{J}} L[\mathcal{J}]$ with $\mathcal{D}_\mathbf{0}^{(b)}$ and $\mathcal{D}_{\boldsymbol{\nu}}^{(b)}$ using CART or TAO
  for all $j \in \mathcal{J}^{(b)}$ **do**
    $\sigma_{j,\mathbf{0}} \leftarrow \sum_{(\boldsymbol{x}_i, w_i) \in \mathcal{D}_\mathbf{0} \cap j} w_i$
    $\sigma_{j,\boldsymbol{\nu}} \leftarrow \sum_{(\boldsymbol{x}_i, w_i) \in \mathcal{D}_{\boldsymbol{\nu}}^{(b)} \cap j} w_i \quad$ for all $\quad \boldsymbol{\nu} \in \mathcal{V}$
    $\hat{\Delta}_{A,j}^{(b)} \leftarrow \left[\sum_{\nu\in\mathcal{V}} \boldsymbol{\nu}\boldsymbol{\nu}^{\mathrm{T}} \right]_{AB}^{-1} \left[\sum_{\nu\in\mathcal{V}} \boldsymbol{\nu} \log \frac{\sigma_{j,\boldsymbol{\nu}}}{\sigma_{j,\mathbf{0}}} \right]_B$
  end for
  $t_{\boldsymbol{\nu}}^{(b)*}(\boldsymbol{x}) \leftarrow \sum_{j\in\mathcal{J}^{(b)}} \mathbb{1}_j(\boldsymbol{x}) \left(\nu_A \hat{\Delta}_{A,j}^{(b)} \right)$
  $\hat{T}_{\boldsymbol{\nu}}^{(b)}(\boldsymbol{x}) \leftarrow \hat{T}^{(b-1)}(\boldsymbol{x}) + \eta^{(b)} t^{(b)*}(\boldsymbol{x})$
end for
return $\hat{T}(\boldsymbol{x}|\boldsymbol{\nu}) = \sum_{b=1}^{B} \eta^{(b)} \sum_{j\in\mathcal{J}^{(b)}} \mathbb{1}_j(\boldsymbol{x}) \nu_A \hat{\Delta}_{A,j}^{(b)}$

For the nominal $\mathbf{0}$ and each $\boldsymbol{\nu} \in \mathcal{V}$. The notation $\mathcal{D}_{\boldsymbol{\nu}}^{(b)} \cap \Delta\boldsymbol{x}_j$ indicates summing over events from $\mathcal{D}_{\boldsymbol{\nu}}^{(b)}$ that fall within the phase-space region $\Delta\boldsymbol{x}_j$ of terminal node j. These estimates, valid for $\boldsymbol{\nu} = \mathbf{0}$ and $\boldsymbol{\nu} \in \mathcal{V}$, yield the new polynomial interpolation at iteration b, with coefficients

$$\hat{\Delta}_{A,j}^{(b)} = \left[\sum_{\nu\in\mathcal{V}} \boldsymbol{\nu}\boldsymbol{\nu}^{\mathrm{T}} \right]_{AB}^{-1} \left[\sum_{\boldsymbol{\nu}\in\mathcal{V}} \boldsymbol{\nu} \log \frac{\sigma_{j,\boldsymbol{\nu}}^{(b)}}{\sigma_{j,\mathbf{0}}^{(b)}} \right]_B . \tag{61}$$

The inverse matrix in the first factor exists if $\mathcal{V}$ has a full-rank coordinate matrix, so we need at least as many training samples $\boldsymbol{\nu} \in \mathcal{V}$ as there are coefficient functions $\hat{\Delta}_A(\boldsymbol{x})$. This linear relation of log-ratios makes equation (61) highly efficient to evaluate. We can then use the CART or 'Tree Alternate Optimization' (TAO) [55–58] algorithms to determine the optimal phase-space partitioning $\mathcal{J}^{(b)}$, completing iteration b. After B boosting iterations, all constants in equation (57) are determined, giving the final DCR estimate

$$\hat{S}(\boldsymbol{x}|\boldsymbol{\nu}) = \exp\left(\nu_A \hat{\Delta}_A(\boldsymbol{x}) \right) \simeq \frac{\mathrm{d}\sigma(\boldsymbol{x}|\boldsymbol{\nu})}{\mathrm{d}\sigma(\boldsymbol{x}|\mathrm{SM})}. \tag{62}$$

It is fast to evaluate, parametric in $\boldsymbol{\nu}$, satisfies $\hat{S}(\boldsymbol{x}|\mathbf{0}) = 1$, and is continuous in both $\boldsymbol{x}$ and $\boldsymbol{\nu}$. Algorithm 1 provides a pseudo-code summary of the steps, defining the BPT algorithm. Appendix C contains a detailed derivation and a simple analytic toy example.

6. Gradually refinable modeling

With the setup for the training of parametric regressors in place, we discuss gradually refinable modeling as a flexible approach to unbinned analyses, incorporating incremental improvements while preserving existing results. Based on procedures commonly employed in binned analyses, we discuss the factorization of POI and nuisance parameter dependencies in the unbinned case. Uncorrelated groups of systematic effect are isolated and subsequently learned independently. With the help of an additive model, summing over the various contributing processes, we can incrementally extend and refine an existing model without invalidating existing components. In this way, gradually refinable modeling aligns with established practices for systematic uncertainty management in binned analyses, extending these strategies to unbinned data while allowing the analysis to evolve with growing data and improved modeling techniques.

6.1. The binned Poisson likelihood

We begin with the binned Poisson likelihood for several observations (N_{bin}) in disjoint phase-space regions (bins), a setup described in detail in [59, 60]. Multiple processes, labeled by $p = 1, \ldots, N_p$, contribute to the

cross-section component $\sigma_{n,p}(\mathrm{SM})$ in bin n. The dependence on SMEFT POIs is assumed to factorize from the systematic effects. Since SMEFT ME-squared terms are polynomial or can be truncated to polynomial form, a small set of non-zero values of $\boldsymbol{\theta}$ suffices to determine the coefficients in the SMEFT parametrization $\sigma_{n,p}(\theta) = R_{n,p}(\boldsymbol{\theta})\sigma_{n,p}(\mathrm{SM})$ [7]. Here, $R_{n,p}(\boldsymbol{\theta})$ satisfies $R_{n,p}(\mathrm{SM}) = 1$ and fully encodes the SMEFT dependence in each bin. If $R_{n,p}(\boldsymbol{\theta}) = 1$ holds to a good approximation for all n, we call the process p a background.

The Poisson expectation of the yield in bin n can then be expressed as

$$\lambda_n(\boldsymbol{\theta},\boldsymbol{\nu}) = \mathcal{L}(\boldsymbol{\nu}) \sum_{p=1}^{N_p} R_{n,p}(\boldsymbol{\theta}) \exp\left(\boldsymbol{\nu}^\intercal \Delta_{n,p,1} + \nu^\intercal \Delta_{n,p,2} \boldsymbol{\nu}\right) \sigma_{n,p}(\mathrm{SM}), \tag{63}$$

where the exponential is a second-order interpolation[4] of the systematic effects in terms of K-dimensional vectors $\Delta_{n,p,1}$ and $K \times K$ matrices $\Delta_{n,p,2}$. The nuisances $\boldsymbol{\nu}$ are conventionally chosen to minimize their linear correlation. In the uncorrelated case, off-diagonal entries in $\Delta_{n,p,2}$ vanish. Small linear nuisance parameter correlations can be accounted for in the penalty [59].

The binned ansatz in equation (63) reflects inductive bias. First, systematic effects are modeled in a factorized form, meaning the constants $\Delta_{n,p,1/2}$ are assumed to be accurately determined by individually varied simulations, with all other model parameters held fixed. Second, the model is additive. This inconspicuous fact, combined with the factorization of systematic effects, is key to enabling gradual model refinement, an implicit feature in the binned case. Once the per-process expectations and systematic parametrizations in equation (63) are established, most of these values can remain unchanged even if the model is refined to include a new process or nuisance parameter. Although this computational saving is modest in the binned case, the unbinned analysis replaces the bin-by-bin constants in equation (63) with $\boldsymbol{x}$-dependent ML parametrizations. Selecting a flexible additive model minimizes the need for re-training when refining the model, offering potentially significant gains in computational efficiency.

The additivity in equation (63) naturally accommodates per-process nuisances related to normalization uncertainties with two key applications. First, higher-order perturbative corrections ('k-factors'), derived from theoretical predictions, enhance the accuracy of inclusive parton-level predictions. These k-factors generally apply to a single process, with reduced uncertainties best captured by nuisances that scale only this component. Second, normalization nuisances are useful for small backgrounds where the pdf in $\mathcal{D}$ can be estimated from $\mathcal{A}$, but normalization uncertainties remain significant. In this case, a normalization nuisance allows for *in-situ* constraints from $\mathcal{D}$. Specifically, setting $\Delta_{n,p,1} = \log \alpha_{\mathrm{norm},p}$ and $\Delta_{n,p,2} = 0$ for all n results in a scaling of process p, where $\alpha_{\mathrm{norm},p}$ is a positive constant that normalizes the impact of the nuisance $\nu_{\mathrm{norm},p}$. Additionally, we can omit $\nu_{\mathrm{norm},p}$ from the penalty, allowing the process's normalization to float during profiling.

6.2. Approximate factorization of systematic effects

We now substitute the DCR in equation (6) with an ML surrogate model. 'Likelihood-free' inference refers to techniques that rely on parametrically evaluating ratios of the extended likelihood, and thus ratios of the differential cross-section $\mathrm{d}\Sigma(\boldsymbol{\theta},\boldsymbol{\nu})$. These alone are enough to evaluate equation (10).

Constructing a generic ML surrogate starts by expressing the unbinned model $\mathrm{d}\Sigma(\boldsymbol{\theta},\boldsymbol{\nu})$ as a sum over weighted sub-processes, with normalization uncertainties treated separately[5]. Nuisance parameters $\nu_{p,\mathrm{norm}}$ are introduced for this purpose. We have

$$\mathrm{d}\Sigma(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\nu}) = \sum_p \alpha_{\mathrm{norm},p}^{\nu_{\mathrm{norm},p}} \, \mathrm{d}\sigma_p(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\nu}), \tag{64}$$

where event samples for each component $\mathrm{d}\sigma_p(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\nu})$ can be obtained from equations (23), (30), or (31). Next, we factorize systematic effects and POI dependence. The SM point is at $\boldsymbol{\theta} = \boldsymbol{\nu} = \mathbf{0}$, and for each $\mathrm{d}\sigma_p(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\nu})$ we have

[4] A detailed account of the options for interpolating binned yields is provided in [21].

[5] The reason for the separate treatment of normalization nuisances is best seen in comparing the Taylor expansion in $\boldsymbol{\nu}$ with the corresponding expansion of the purely multiplicative model in [17] where nuisances are modeled relative to the total differential cross-section instead of per-process. For arbitrary values of normalization nuisances, a polynomial expansion of the logarithm of the *total* differential cross-section requires arbitrarily many terms that would have to be learned individually. The ansatz in equation (64) will reduce the ensuing ML task to a straightforward classification problem, one for each process.

$$\frac{\mathrm{d}\sigma_p(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\nu})}{\mathrm{d}\sigma_p(\boldsymbol{x}|\boldsymbol{0},\boldsymbol{0})} = \frac{\mathrm{d}\sigma_p(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\nu})}{\mathrm{d}\sigma_p(\boldsymbol{x}|\boldsymbol{0},\boldsymbol{\nu})}\frac{\mathrm{d}\sigma_p(\boldsymbol{x}|\boldsymbol{0},\boldsymbol{\nu})}{\mathrm{d}\sigma_p(\boldsymbol{x}|\boldsymbol{0},\boldsymbol{0})} \approx \frac{\mathrm{d}\sigma_p(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{0})}{\mathrm{d}\sigma_p(\boldsymbol{x}|\boldsymbol{0},\boldsymbol{0})}\frac{\mathrm{d}\sigma_p(\boldsymbol{x}|\boldsymbol{0},\boldsymbol{\nu})}{\mathrm{d}\sigma_p(\boldsymbol{x}|\boldsymbol{0},\boldsymbol{0})} \equiv \underbrace{\frac{\mathrm{d}\sigma_p(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{0})}{\mathrm{d}\sigma_p(\boldsymbol{x}|\mathrm{SM})}}_{\hat{R}_p(\boldsymbol{x}|\boldsymbol{\theta})}\underbrace{\frac{\mathrm{d}\sigma_p(\boldsymbol{x}|\boldsymbol{0},\boldsymbol{\nu})}{\mathrm{d}\sigma_p(\boldsymbol{x}|\mathrm{SM})}}_{\hat{S}_p(\boldsymbol{x}|\boldsymbol{\nu})}. \tag{65}$$

This factorization works if SMEFT effects are independent of systematic effects, i.e.

$$\frac{\mathrm{d}\sigma_p(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\nu})}{\mathrm{d}\sigma_p(\boldsymbol{x}|\boldsymbol{0},\boldsymbol{\nu})} \approx \frac{\mathrm{d}\sigma_p(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{0})}{\mathrm{d}\sigma_p(\boldsymbol{x}|\boldsymbol{0},\boldsymbol{0})}. \tag{66}$$

The factor

$$\hat{R}_p(\boldsymbol{x}|\boldsymbol{\theta}) \simeq \frac{\mathrm{d}\sigma_p(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{0})}{\mathrm{d}\sigma_p(\boldsymbol{x}|\mathrm{SM})}. \tag{67}$$

Approximates SMEFT variations as a polynomial in $\boldsymbol{\theta}$ and can be obtained from methods in [8–17]. Systematic effects are parametrized by

$$\hat{S}_p(\boldsymbol{x}|\boldsymbol{\nu}) \simeq \frac{\mathrm{d}\sigma_p(\boldsymbol{x}|\boldsymbol{0},\boldsymbol{\nu})}{\mathrm{d}\sigma_p(\boldsymbol{x}|\mathrm{SM})}. \tag{68}$$

We can learn this parametric dependence, one effect at a time, using the strategy in section 5. The ML model can be a neural network or the tree-based algorithm from section 5.4.

The validity of equation (66) must be established on a case-by-case basis and can be verified through simulation. The separation of particle and detector-level effects from SMEFT effects is generally accurate due to the different energy scales involved; the POIs typically do not influence low-energy detector interactions. At the parton level, we need to verify the independence of POIs from systematic effects. PDFs, for example, may depend on SMEFT POIs [61], so this correlation should not be neglected without careful consideration. Similarly, the linear and quadratic SMEFT terms may have scale uncertainties differing from the SM prediction. In this case, $\hat{S}_p(\boldsymbol{x}|\nu_\mathrm{R},\nu_\mathrm{F})$ should be trained with synthetic scale variations that cover scale variations for non-zero POIs. A suitably flexible model should accommodate these subtle analysis-dependent effects, which we leave to future treatment. From now on, we assume the factorization

$$\hat{R}_p(\boldsymbol{x}|\boldsymbol{\theta})\,\hat{S}_p(\boldsymbol{x}|\boldsymbol{\nu}) \simeq \frac{\mathrm{d}\sigma_p(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\nu})}{\mathrm{d}\sigma_p(\boldsymbol{x}|\mathrm{SM})}. \tag{69}$$

Holds accurately. Following the same steps, we factorize $\hat{S}_p(\boldsymbol{x}|\boldsymbol{\nu})$ into uncorrelated groups of systematic uncertainties and train each factor with equation (53). For instance, uncorrelated one-parameter systematic uncertainties with quadratic accuracy simplify the surrogate to

$$\hat{S}_p(\boldsymbol{x}|\boldsymbol{\nu}) = \prod_{k=1}^{K} \exp\left(\nu_k \hat{\Delta}_{p,k,1}(\boldsymbol{x}) + \nu_k^2 \hat{\Delta}_{p,k,2}(\boldsymbol{x})\right). \tag{70}$$

With $2K$ real-valued functions $\hat{\Delta}_{p,k,1}(\boldsymbol{x})$ and $\hat{\Delta}_{p,k,2}(\boldsymbol{x})$ for each p. In most cases, first or second-degree polynomials are sufficient, though the method allows higher degrees.

6.3. A general unbinned surrogate model

In analogy to equation (63), we define a general model for the fiducial differential cross-section,

$$\mathrm{d}\Sigma(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\nu}) = \sum_{p=1}^{N_p} \hat{R}_p(\boldsymbol{x}|\boldsymbol{\theta})\,\alpha_{\mathrm{norm},p}^{\nu_{\mathrm{norm},p}}\,\hat{S}_p(\boldsymbol{x}|\boldsymbol{\nu})\,\mathrm{d}\sigma_p(\boldsymbol{x}|\mathrm{SM}). \tag{71}$$

Next, we create a likelihood-free ML surrogate, relying only on differential DCRs. Dividing by

$$\mathrm{d}\Sigma(\boldsymbol{x}|\mathrm{SM}) = \sum_p \mathrm{d}\sigma_p(\boldsymbol{x}|\mathrm{SM}). \tag{72}$$

Gives per-process DCRs that we replace with surrogates. The simplest approach divides each $\mathrm{d}\sigma_p(\boldsymbol{x}|\mathrm{SM})$ in equation (71) by equation (72),

$$\frac{\mathrm{d}\Sigma(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\nu})}{\mathrm{d}\Sigma(\boldsymbol{x}|\mathrm{SM})} = \sum_{p=1}^{N_p} \hat{R}_p(\boldsymbol{x}|\boldsymbol{\theta})\, \alpha_{\mathrm{norm},p}^{\nu_{\mathrm{norm},p}}\, \hat{S}_p(\boldsymbol{x}|\boldsymbol{\nu})\, \hat{g}_p(\boldsymbol{x}) \quad \text{where} \quad \hat{g}_p(\boldsymbol{x}) \simeq \frac{\mathrm{d}\sigma_p(\boldsymbol{x}|\mathrm{SM})}{\sum_q \mathrm{d}\sigma_q(\boldsymbol{x}|\mathrm{SM})}. \tag{73}$$

Estimates the DCR of each process relative to the SM total.

A classifier trained to distinguish process p from the total SM simulation can learn $\hat{g}_p(\boldsymbol{x})$ using the likelihood ratio trick. The DCR for arbitrary denominators, as required for profiling in equation (1), can then be obtained from double ratios, allowing $\mathrm{d}\Sigma_p(\boldsymbol{x}|\mathrm{SM})$ to cancel out.

Equation (73) provides significant modeling flexibility, as the quotient of equations (71) and (72) can be represented in various ways using the surrogates $\hat{g}_p(\boldsymbol{x})$. We present two examples demonstrating how this flexibility can solve common challenges in analysis development.

6.4. Refining an existing model

New systematic effects can be gradually incorporated, as expanding the dimension of the nuisance vector $\boldsymbol{\nu}$ in, for example, equation (70) does not invalidate existing surrogates $\hat{\Delta}_{p,1/2}$. Only the new components require training.

Similarly, additional background sources can be seamlessly included. With the additive structure of equation (71), a new process can be added by adjusting the steps leading to equation (73). For instance, if a missing background $\mathrm{d}\sigma_{\mathrm{BKG}}(\boldsymbol{x})$, with $R_p(\boldsymbol{x}|\boldsymbol{\theta}) = 1$, is identified, we can extend $\mathrm{d}\Sigma(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\nu})$ by adding a term,

$$\mathrm{d}\Sigma'(\boldsymbol{x}|\boldsymbol{\nu},\boldsymbol{\theta}) = \mathrm{d}\Sigma(\boldsymbol{x}|\boldsymbol{\nu},\boldsymbol{\theta}) + \mathrm{d}\sigma_{\mathrm{BKG}}(\boldsymbol{x}). \tag{74}$$

We can express the DCR in terms of the existing model as

$$\begin{aligned}\frac{\mathrm{d}\Sigma'(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\nu})}{\mathrm{d}\Sigma'(\boldsymbol{x}|\mathrm{SM})} &= \frac{\mathrm{d}\Sigma(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\nu}) + \mathrm{d}\sigma_{\mathrm{BKG}}(\boldsymbol{x})}{\mathrm{d}\Sigma(\boldsymbol{x}|\mathrm{SM}) + \mathrm{d}\sigma_{\mathrm{BKG}}(\boldsymbol{x})} = \frac{\frac{\mathrm{d}\Sigma(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\nu})}{\mathrm{d}\Sigma(\boldsymbol{x}|\mathrm{SM})} + \frac{\mathrm{d}\sigma_{\mathrm{BKG}}(\boldsymbol{x})}{\mathrm{d}\Sigma(\boldsymbol{x}|\mathrm{SM})}}{1 + \frac{\mathrm{d}\sigma_{\mathrm{BKG}}(\boldsymbol{x})}{\mathrm{d}\Sigma(\boldsymbol{x}|\mathrm{SM})}} \\ &\simeq \frac{\sum_{p=1}^{N_p} \hat{R}_p(\boldsymbol{x}|\boldsymbol{\theta})\, \alpha_{\mathrm{norm},p}^{\nu_{\mathrm{norm},p}}\, \hat{S}_p(\boldsymbol{x}|\boldsymbol{\nu})\, \hat{g}_p(\boldsymbol{x}) + \hat{g}'(\boldsymbol{x})}{1 + \hat{g}'(\boldsymbol{x})}\end{aligned} \tag{75}$$

where

$$\hat{g}'_p(\boldsymbol{x}) \simeq \frac{\mathrm{d}\sigma_{\mathrm{BKG}}(\boldsymbol{x})}{\sum_q \mathrm{d}\sigma_q(\boldsymbol{x}|\mathrm{SM})}. \tag{76}$$

The only new component is $\hat{g}'(\boldsymbol{x})$, a classifier that gives the DCR for the new process relative to the previous total. The new process adds to both the numerator and denominator, with no change to the rest of the model. If the new background has uncertainties, we replace $\mathrm{d}\sigma_{\mathrm{BKG}}(\boldsymbol{x})$ with $\mathrm{d}\sigma_{\mathrm{BKG}}(\boldsymbol{x}|\boldsymbol{\nu})$ in equation (74) and repeat the derivation, yielding

$$\frac{\mathrm{d}\Sigma'(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\nu})}{\mathrm{d}\Sigma'(\boldsymbol{x}|\mathrm{SM})} \simeq \frac{\sum_{p=1}^{N_p} \hat{R}_p(\boldsymbol{x}|\boldsymbol{\theta})\, \alpha_{\mathrm{norm},p}^{\nu_{\mathrm{norm},p}}\, \hat{S}_p(\boldsymbol{x}|\boldsymbol{\nu})\, \hat{g}_p(\boldsymbol{x}) + \hat{S}_{\mathrm{BKG}}(\boldsymbol{x}|\boldsymbol{\nu})\, \hat{g}'(\boldsymbol{x})}{1 + \hat{g}'(\boldsymbol{x})}, \tag{77}$$

where $g'(\boldsymbol{x})$ from equation (76) remains, and we only need to learn one extra factor, $\hat{S}_{\mathrm{BKG}}(\boldsymbol{x}|\boldsymbol{\nu})$, to model the background's systematic effects,

$$\hat{S}_{\mathrm{BKG}}(\boldsymbol{x}|\boldsymbol{\nu}) \simeq \frac{\mathrm{d}\sigma_{\mathrm{BKG}}(\boldsymbol{x}|\boldsymbol{0},\boldsymbol{\nu})}{\mathrm{d}\sigma_{\mathrm{BKG}}(\boldsymbol{x}|\mathrm{SM})}. \tag{78}$$

Similar to equation (68). Refinable modeling thus avoids retraining existing regressors and enables incremental analysis development. Because an event sample for $\mathrm{d}\sigma_{\mathrm{BKG}}$ is the only ingredient for obtaining $\hat{g}'(\boldsymbol{x})$, it could alternatively be measured in real-data side bands, supporting the development of data-driven unbinned estimation strategies.

6.5. Refinement for a high-purity process

Now, consider a single SMEFT-dependent signal process $\mathrm{d}\sigma_{\mathrm{SMEFT}}(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\nu})$ and multiple $\boldsymbol{\theta}$-independent backgrounds, $\mathrm{d}\sigma_p(\boldsymbol{x}|\boldsymbol{\nu})$. To form the DCR, we divide each term in the sums of both sides of equation (73) by $\mathrm{d}\sigma_{\mathrm{SMEFT}}(\boldsymbol{x}|\mathrm{SM})$, which leaves the result unchanged but modifies the parametrization to

$$\frac{\mathrm{d}\Sigma(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\nu})}{\mathrm{d}\Sigma(\boldsymbol{x}|\mathrm{SM})} \simeq \frac{\alpha_{\mathrm{norm}}^{\nu_{\mathrm{norm}}}\,\hat{R}(\boldsymbol{x}|\boldsymbol{\theta})\;\hat{S}(\boldsymbol{x}|\boldsymbol{\nu}) + \sum_{p=1}^{N_p}\alpha_{\mathrm{norm},p}^{\nu_{\mathrm{norm},p}}\,\hat{S}_p(\boldsymbol{x}|\boldsymbol{\nu})\,\hat{g}_p''(\boldsymbol{x})}{1+\sum_p \hat{g}_p''(\boldsymbol{x})} \tag{79}$$

where

$$g_p''(\boldsymbol{x}) \simeq \frac{\mathrm{d}\sigma_p(\boldsymbol{x}|\mathrm{SM})}{\mathrm{d}\sigma_{\mathrm{SMEFT}}(\boldsymbol{x}|\mathrm{SM})}. \tag{80}$$

And signal quantities have no process index. The classifier $\hat{g}''(\boldsymbol{x})$ is trained to distinguish each background from the SMEFT signal at the SM point. Adding a new background process only requires training one additional classifier, as expanding the sum over p leaves the rest of the model unaffected.

The steps in this and the previous section can be combined and repeated as modeling is refined. Equation (71) thus supports incremental refinements, similar to the binned model in equation (63). Once a new effect prediction is available, it can be incorporated. Unlike in the binned case, expectations here come from separately trained surrogates rather than binned yields.

7. Top quark pair production in the 2ℓ channel

As an example, we study dileptonic top quark pair production in pp collisions at $\sqrt{s} = 13\,\mathrm{TeV}$, $\mathrm{pp} \to \boldsymbol{t\bar{t}} \to \mathrm{b}\ell^+\nu_\ell\bar{\mathrm{b}}\ell^-\bar{\nu}_\ell$, or $\boldsymbol{t\bar{t}}(2\ell)$ for short. The event simulators provide all necessary quantities at the parton and particle levels. For calibration of reconstructed objects (jets, missing transverse momentum, and leptons), ATLAS and CMS open data projects [62, 63] supply uncertainty information. This is enough to demonstrate the tools' application. A fully realistic detector simulation with all data-dependent systematic effects is neither feasible nor needed. We focus on a heuristic treatment of the main uncertainties in the differential cross-section. A detailed binned measurement of $\boldsymbol{t\bar{t}}(2\ell)$, including a full account of systematic uncertainties, is available from ATLAS [64] and CMS [65]. We focus on a heuristic treatment of the main uncertainties in the differential cross-section. A detailed binned measurement of $\boldsymbol{t\bar{t}}(2\ell)$, including a full account of systematic uncertainties, is available from ATLAS [64] and CMS [65].

7.1. Event simulation

We generate the $\boldsymbol{t\bar{t}}(2\ell)$ signal process with MadGraph5_aMC@NLO v2.6.5 [31] at leading order and use the NNPDF PDFs v3.1 [66]. We simulate the top quark pairs at $\sqrt{s} = 13\,\mathrm{TeV}$, followed by leptonic decays of the W bosons ($\ell = e, \mu, \tau$), and employ the SMEFTsim v3.0 model [67] for simulating the parton-level SMEFT effects. The ME simulation is interfaced to PYTHIA v8.226 [68] using the CP5 tune [69, 70] for fragmentation, PS, and hadronization of partons in the initial and final states, along with the underlying event and multiparton interactions. The ME for the $\boldsymbol{t\bar{t}}$ signal includes up to one extra parton. Double counting of the partons generated with MadGraph5_aMC@NLO and PYTHIA is removed using the MLM [71] scheme. The events are subsequently processed with a Delphes-based simulation model of the CMS detector [44]. Kinematic requirements are placed on jets, electrons, and muons. Jets are reconstructed with anti-k_T algorithm [72] using a distance parameter of 0.4 in the FastJet software package [73]. The nominal b tagging of jets in Delphes is based on parton-matching and a parametrization of the nominal CMS b-tagging efficiency. Electrons and muons must be isolated from jets, satisfy $p_{\mathrm{T}} > 20\,\mathrm{GeV}$, and be reconstructed within absolute pseudorapidity $|\eta| < 2.5$. If there are two same-flavor lepton candidates of opposite electric charge within a 10 GeV window around the Z boson mass, $|m(\ell^+\ell^-) - m_Z| < 10\,\mathrm{GeV}$, the event is rejected. According to [65], the purity after the Z boson mass veto is 95%, with a small background from the Drell–Yan process. We ignore the contribution from Drell–Yan in the following. Jets must satisfy $p_{\mathrm{T}} > 30\,\mathrm{GeV}$ and $|\eta| < 2.4$, and there must be more than two jets, among which at least two must be b tagged.

Using the Delphes objects, we reconstruct the top quark kinematic quantities described in [65]. This provides access to SMEFT-sensitive observables, including the top quarks' invariant masses, angles, and transverse momenta. To reduce the computational demand while keeping sensitivity to SMEFT effects, we require $m(\boldsymbol{t\bar{t}}) > 750\,\mathrm{GeV}$, corresponding to an inclusive fiducial cross-section of 0.31 pb [65]. We normalize the Delphes simulation of a total of 1.2×10^6 events to this value and use a central value for the integrated luminosity $\mathcal{L}_0 = 137\,\mathrm{fb}^{-1}$ with a conservative 5% log-normal uncertainty,

$$\mathcal{L}(\boldsymbol{\nu}) = \mathcal{L}_0\,\alpha_{\mathrm{lumi}}^{\nu_{\mathrm{lumi}}}. \tag{81}$$

We simulate the effects from the real and imaginary part of the Wilson coefficient C_{tG}, and the four-fermion operators $C_{Qq}^{(1,8)}$, $C_{Qq}^{(3,8)}$, and $C_{qt}^{(8)}$. Our five POIs are, therefore, the Wilson coefficients `ctGRe`, `ctGIm`, `cQj18`, `cQj38`, and `ctj8` in the conventions of [67], multiplying the operators

$$O_{tG} = (\bar{Q}\sigma^{\mu\nu}T^a t)\,\tilde{H}G^a_{\mu\nu}, \qquad O_{Qq}^{(3,8)} = \left(\bar{Q}\sigma^i T^a\gamma_\mu Q\right)\left(\bar{q}\sigma^i T^a\gamma^\mu q\right),$$
$$O_{Qq}^{(1,8)} = (\bar{Q}T^a\gamma_\mu Q)\,(\bar{q}T^a\gamma^\mu q), \qquad O_{tj}^{(8)} = (\bar{t}T^a\gamma_\mu t)\,(\bar{u}T^a\gamma^\mu u),$$

where the left and right-chiral lower-case quark fields q and u belong to the first and second generation. The third-generation left-chiral quark doublet is denoted by Q. A non-zero value of C_{tG} provides CP-even and CP-odd modifications to the top-gluon interaction. The four fermion operators add contact interactions of the first and second with the third-generation quark currents. For obtaining the SMEFT predictions, we use the reweighting technique discussed in section 4.4.3 and [7]. The dominant effect of these operators on the $\boldsymbol{t\bar{t}}(2\ell)$ cross-section is linear with the Wilson coefficients [74] so that complications from dominantly quadratic predictions that violate Wilk's theorem when computing the distribution of the profile likelihood test statistic can be neglected [27].

The following event-level features define the observation $\boldsymbol{x}$. From the reconstructed top quark momenta, we compute the invariant mass $m(\boldsymbol{t\bar{t}})$, the transverse momentum $p_{\mathrm{T}}(\boldsymbol{t\bar{t}})$, the rapidity difference $\Delta\eta(\boldsymbol{t\bar{t}}) = \eta(\mathrm{t}) - \eta(\bar{\mathrm{t}})$ and the difference of absolute rapidities of the top and anti-top quark, $\Delta|\eta|(\boldsymbol{t\bar{t}}) = |\eta(\mathrm{t})| - |\eta(\bar{\mathrm{t}})|$. The quantities $m(\boldsymbol{t\bar{t}})$ and $p_{\mathrm{T}}(\boldsymbol{t\bar{t}})$ are sensitive to SMEFT effects with energy-growth while $\Delta|\eta|(\boldsymbol{t\bar{t}})$ is sensitive to the effects of the charge-asymmetry [75, 76], modified, e.g. by $C_{tj}^{(8)}$. Furthermore, we include the transverse momentum and the pseudo-rapidity of the top and anti-top quark. Because leptons are clean probes of the possible SMEFT effects, independent of the hadronic activity, we also include the invariant mass $m(\ell^+\ell^-)$, the transverse momentum $p_{\mathrm{T}}(\ell^+\ell^-)$, the rapidity difference $\Delta\eta(\ell^+\ell^-)$, and the difference of absolute rapidities $\Delta|\eta|(\ell^+\ell^-)$ of the dilepton system. Finally, we include the absolute value of the difference of the azimuthal lepton angles $|\Delta\varphi_{\mathrm{lab}}|(\ell^+\ell^-)$ and the cosine of the spatial angle between the leptons $\cos(\phi_{\mathrm{lab}}(\ell^+\ell^-))$ as measured in the lab frame. The distributions of these observables for the SM and non-zero values for the Wilson coefficients are shown in figure 4. We find good agreement with the study in [8]. The CMS measurement of the spin-correlation in $\boldsymbol{t\bar{t}}$ [77] found constraining power for C_{tG} in the distribution of products of angular observables of the leptons, measured in a specific reference frame spanned by the top quark momentum and the beam plane [78]. These variables characterize the spin-density matrix of the $\boldsymbol{t\bar{t}}(2\ell)$ system, and we present a brief description and their distributions in appendix D. The resulting distributions for non-zero values of the Wilson coefficients are shown in figure 4.

The estimate of the detector-level SMEFT dependence of the signal process $\hat{R}(\boldsymbol{x}|\boldsymbol{\theta})$ can be learned by one of the tools in [8–17]. We use the BIT technique [9, 10] to learn the polynomial dependence up to the quadratic order. Concretely, we train trees with a maximum depth of four in $B = 300$ boosting iterations with a learning rate of $\eta = 0.2$. We regularize each tree by requiring at least 50 events in each node.

We use the parametric tree from section 5.4 to estimate the systematic effects discussed in the following sections. Similar settings turn out to be almost universally applicable. We keep the maximum tree depth at four in all cases and use $B = 300$ boosting iterations and a learning rate of $\eta = 0.2$. When we obtain the synthetic data from reweighting, such that there are no independent stochastic fluctuations in the various terms in the loss function, a minimum node size requirement of 50 events proves sufficient to regularize the trees. For systematic variations where $\boldsymbol{x}$ changes with $\boldsymbol{\nu}$, we raise this regulator requirement to 500.

In the subsequent sections, we discuss uncertainties in the renormalization and factorization scales (scale), the difference between the MadGraph5_aMC@NLO and the POWHEG event generator (POW), the normalization of the signal process (norm), the jet momentum calibration (JES), the b-tagging efficiency (HF) and light-quark mis-tagging probability (LF), and the lepton efficiency calibration (ℓ). The model, therefore, is given in terms of the DCR

$$\mathcal{R}(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\nu}) \equiv \frac{\mathrm{d}\Sigma(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\nu})}{\mathrm{d}\Sigma(\boldsymbol{x}|\mathrm{SM})} = \alpha_{\mathrm{norm}}^{\nu_{\mathrm{norm}}}\hat{R}(\boldsymbol{x}|\boldsymbol{\theta})\;\hat{S}_{\mathrm{scale}}(\boldsymbol{x}|\nu_R,\nu_{\mathrm{F}})\;\hat{S}_{\mathrm{POW}}(\boldsymbol{x}|\nu_{\mathrm{POW}})\,\hat{S}_{\mathrm{JES}}(\boldsymbol{x}|\nu_{\mathrm{JES}})$$
$$\times\,\hat{S}_{\mathrm{LF}}(\boldsymbol{x}|\nu_{\mathrm{LF}})\,\hat{S}_{\mathrm{HF}}(\boldsymbol{x}|\nu_{\mathrm{HF}})\,\hat{S}_{\ell}(\boldsymbol{x}|\nu_{\ell}). \tag{82}$$

With the individual factors defined in the following.

7.2. Parton-level uncertainties

Among the largest systematic effects are uncertainties in the factorization and renormalization scales, as detailed in section 4.2. Following equation (16), setting $\nu_{\mathrm{R}} = \pm 1$ and $\nu_{\mathrm{F}} = \pm 1$ varies the scales μ_{R} and μ_{F} by a factor of 2. From simulation, we obtain event weights for all eight scale combinations,

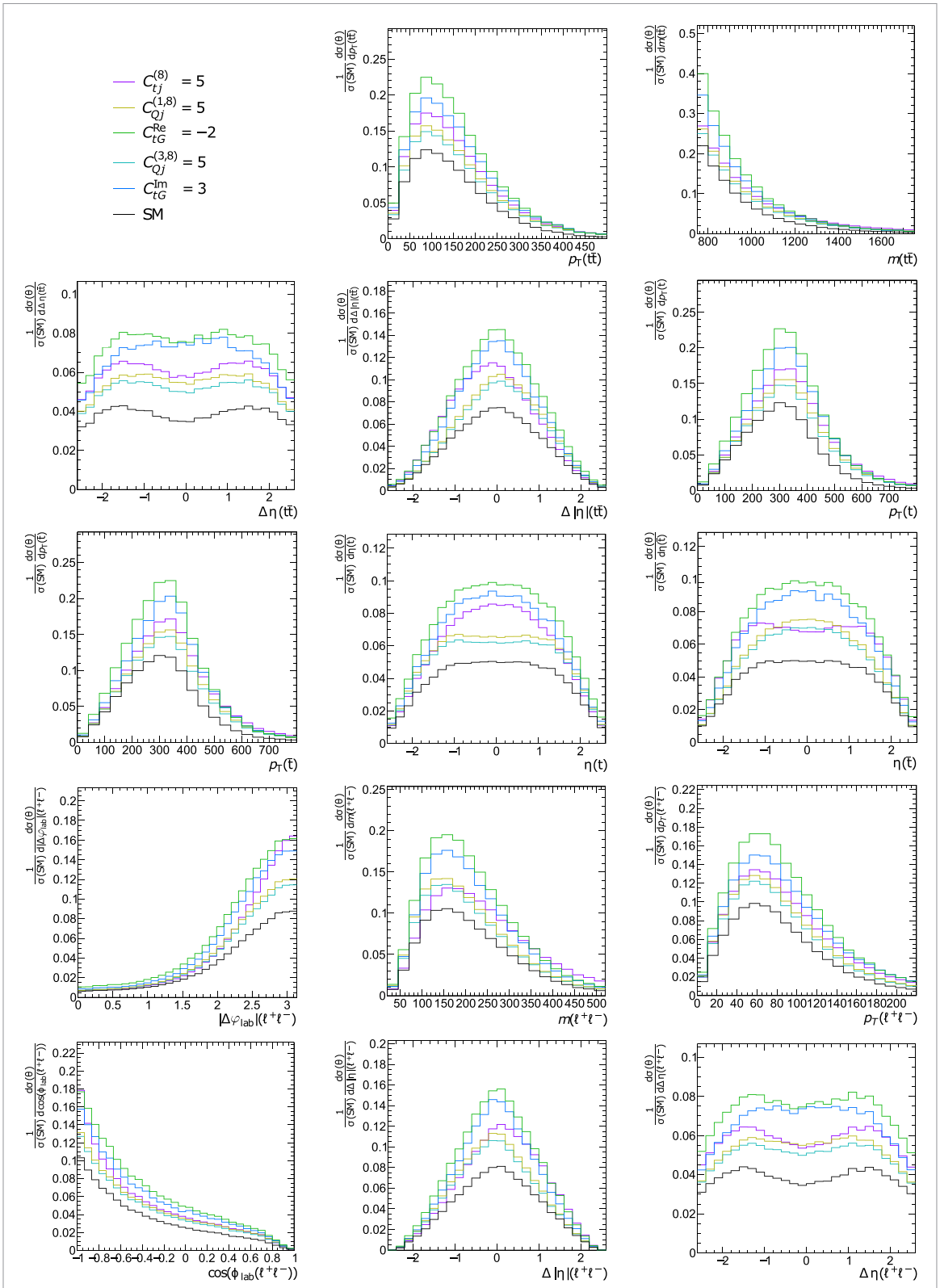

Figure 4. Kinematic features of the $\bar{t}t(2\ell)$ SMEFT simulation for different values of the Wilson coefficients as described in the text.

$$(\nu_R, \nu_F) \in \mathcal{V} = \{(-1,-1), (-1,0), (-1,1), (0,-1), (0,1), (1,-1), (1,0), (1,1)\}. \tag{83}$$

With the nominal SM simulation at $\nu_R = \nu_F = 0$. We model the scale uncertainties up to quadratic accuracy in the nuisance parameters with the ansatz

$$\hat{S}_{\text{scale}}(\boldsymbol{x}|\nu_R, \nu_F) = \exp\left(\nu_R \hat{\Delta}_R(\boldsymbol{x}) + \nu_F \hat{\Delta}_F(\boldsymbol{x}) + \nu_R^2 \hat{\Delta}_{RR}(\boldsymbol{x})(\boldsymbol{x}) + \nu_F^2 \hat{\Delta}_{FF}(\boldsymbol{x}) + \nu_R \nu_F \hat{\Delta}_{RF}(\boldsymbol{x})\right). \tag{84}$$

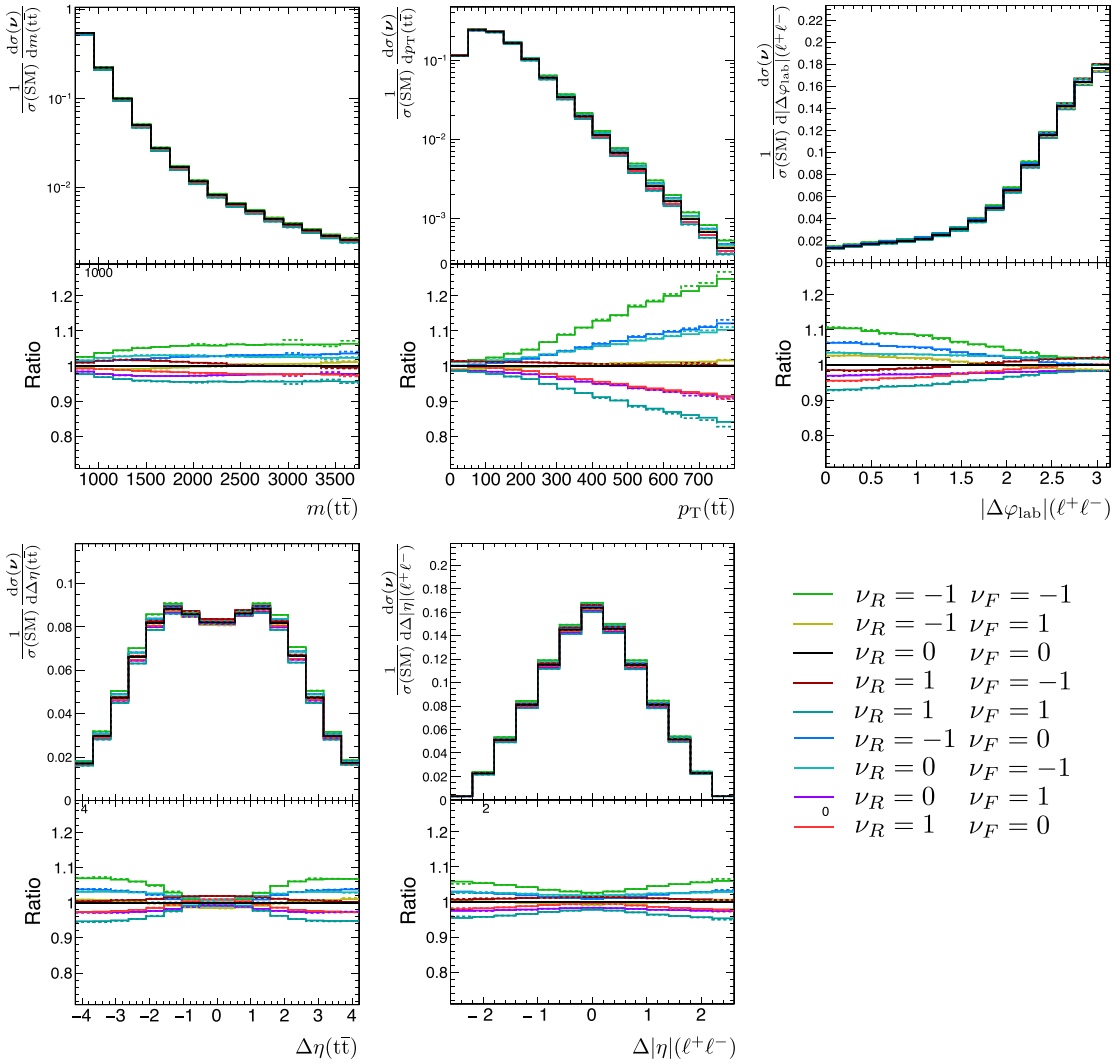

Figure 5. Variations of the simulation of the $\boldsymbol{t\bar{t}}(2\ell)$ simulation with the nuisance parameters ν_R and ν_F, modifying the renormalization and factorization scales, respectively. The dashed lines show the training data, and the solid lines show the prediction from $\hat{s}_\mathrm{scale}(\boldsymbol{x}|\nu_\mathrm{R},\nu_\mathrm{F})$ as obtained from the BPT. All shapes are normalized to the simulation at the SM parameter point, corresponding to $\nu_\mathrm{R} = \nu_\mathrm{F} = 0$.

The five independent terms, labeled by $A = \mathrm{R}, \mathrm{F}, \mathrm{RR}, \mathrm{FF}, \mathrm{RF}$, cover the two linear, two quadratic, and one mixed term. The eight variations in $\mathcal{V}$ thus overconstrain these five functions.

We fit the BPT from section 5.4 in the standard configuration from section 7.1. The result is shown in figure 5 with one-dimensional projections for the observables most sensitive to variations in ν_R and ν_F. Correlated scale variations by a factor of 2 for μ_R and μ_F, corresponding to $\nu_\mathrm{R} = \nu_\mathrm{F} = \pm 1$ reach 8%–10%. The exception is p_T, where the tail shows variations exceeding 20%. In this range, the fit has a small deficit of 1%–2% relative to true variations, likely due to residual inflexibility in the quadratic model. Since μ_R and μ_F capture uncertainties from limited perturbative control, we ignore this slight mismatch for now. Other kinematic features show less shape dependence.

We also simulate events at the SM parameter point using the alternative POWHEG generator, producing a similarly sized event sample for the $\boldsymbol{t\bar{t}}(2\ell)$ process at NLO accuracy in the strong coupling constant. As outlined in section 5.3, and with the caveats noted there, we assign a nuisance parameter ν_POW, with $\nu_\mathrm{POW} = 0$ representing MADGRAPH5_aMC@NLO and $\nu_\mathrm{POW} = 1$ representing POWHEG. Here, $\mathcal{V} = \{1\}$ allows us to train a single-parameter linear surrogate for the (log-) DCR, denoted $\hat{s}_\mathrm{POW}(\boldsymbol{x}|\nu_\mathrm{POW})$. Figure 6 shows one-dimensional projections of the features, revealing shape differences. Minor statistical fluctuations appear in the data tails due to the stochastic independence of the samples, but the BPT averages them out.

Uncertainties in the PDFs, which would require around 100 nuisance parameters for variations along the PDF eigendirections [79], are deferred for future treatment. Instead, and to account for uncertainties in the $m(\boldsymbol{t\bar{t}})$ selection efficiency, we include a normalization uncertainty of 15%, setting $\alpha_\mathrm{norm} = 1.15$.

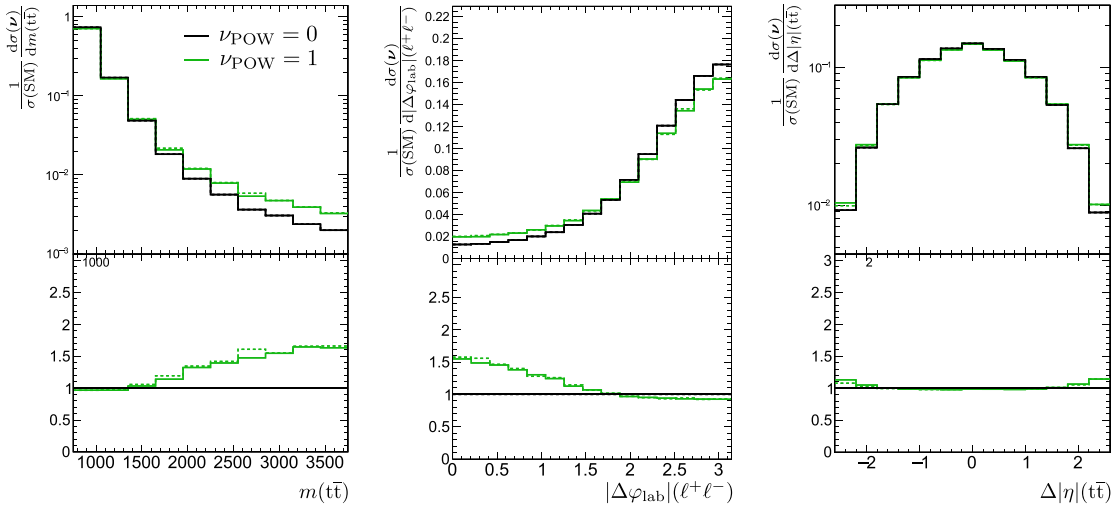

Figure 6. A comparison of MadGraph5_aMC@NLO event simulation (black) and POWHEG event simulation (green, dashed) at the SM parameter point. The solid green line shows the prediction from $\hat{S}_{\mathrm{POW}}(\boldsymbol{x}|\boldsymbol{\nu}_{\mathrm{POW}})$ for the two-point nuisance parameter ν_{POW}.

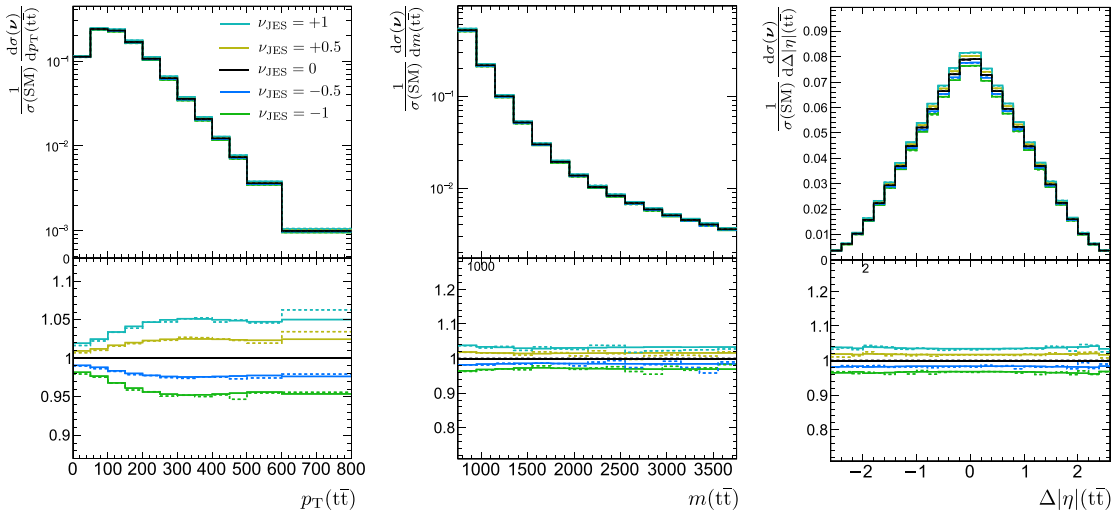

Figure 7. Variations of the nominal simulation with the nuisance parameter for JES uncertainties $\boldsymbol{\nu}_{\mathrm{JES}}$. The dashed lines show the training data and the solid lines show the result from the surrogate $\hat{S}_{\mathrm{JES}}(\boldsymbol{x}|\nu_{\mathrm{JES}})$.

7.3. Jet energy calibration uncertainties

To evaluate the impact of uncertainties in the reconstructed transverse jet momenta, we vary the Delphes-predicted values according to the 'total' CMS JES uncertainty provided in [62]. These variations affect the event selection, missing energy, top quark kinematic reconstruction, and other event features in $\boldsymbol{x}$. Since the per-jet variations depend on the nominal p_{T} and pseudo-rapidity, which are latent (not in $\boldsymbol{x}$), the resulting function $J_{\nu_{\mathrm{JES}}}(\boldsymbol{x}, \boldsymbol{z})$ is also dependent on the latent event configuration.

Using the method in section 4.4.1, we define synthetic data sets and set $\mathcal{V} = \{-1, -0.5, 0.5, 1\}$ for the JES nuisance parameter ν_{JES}, parameterizing the per-jet variation effects on $\boldsymbol{x}$ in units of the JES uncertainty standard deviations. The half-integer values for ν_{JES} provide more granularity than the typical $\pm 1\sigma$ variations used in binned LHC analyses. We then fit a log-linear surrogate,

$$\hat{S}_{\mathrm{JES}}(\boldsymbol{x}|\nu_{\mathrm{JES}}) = \exp\left(\nu_{\mathrm{JES}}\hat{\Delta}_{\mathrm{JES}}(\boldsymbol{x})\right). \tag{85}$$

To model the JES dependence. Figure 7 shows an excellent fit of the surrogate to the variations in the training data. Most observables show a flat variation, except for $p_{\mathrm{T}}(\boldsymbol{t\bar{t}})$, which rises from 2%–5%. In the tails of $m(\boldsymbol{t\bar{t}})$, slight asymmetries in the training data variations are not captured by the linear model, as it approximately symmetrizes the total uncertainty. Refinement with a higher-degree surrogate is left for future work.

7.4. Uncertainties in tagging efficiencies

Uncertainties in the b-tagging efficiency for jets, along with their application, are provided in [62]. This approach relies on p_{T}, pseudo-rapidity, and a nominal binary b-tag label from DELPHES. To apply variations, we also need the generator-level jet flavor f within $\{\mathrm{udsg}, \mathrm{c}, \mathrm{b}\}$. Using the nominal DELPHES simulation, we parametrize the p_{T} and η-dependent b-tagging efficiencies $\varepsilon_f(p_{\mathrm{T}}, \eta)$ for each flavor. Two systematic uncertainties are considered with scale factors $\mathrm{SF}_f(p_{\mathrm{T}}, \eta)$ and variations $\Delta\mathrm{SF}_f(p_{\mathrm{T}}, \eta)$. The HF tagging uncertainty covers the b and c-quark tagging rates, modified in a correlated way for non-zero ν_{HF}. The light-flavor (LF) mistagging uncertainty addresses tagging rates for light-quark and gluon jets, associated with ν_{LF}. The reweighting function for synthetic data in equation (31) is given by

$$r(\boldsymbol{x}_i, \boldsymbol{z}_i|\nu_k, 0) = \frac{F(\nu_k, \text{jets in event } i)}{F(0, \text{jets in event } i)} \tag{86}$$

where

$$\begin{aligned} F(\nu_k, \text{jets}) = &\prod_{\text{tagged jets}} \varepsilon_f(p_{\mathrm{T}}, \eta) \left(\mathrm{SF}_f(p_{\mathrm{T}}, \eta) + \nu_k \Delta\mathrm{SF}_{f,k}(p_{\mathrm{T}}, \eta)\right) \\ &\times \prod_{\text{untagged jets}} \left(1 - \varepsilon_f(p_{\mathrm{T}}, \eta) \left(\mathrm{SF}_f(p_{\mathrm{T}}, \eta) + \nu_k \Delta\mathrm{SF}_{f,k}(p_{\mathrm{T}}, \eta)\right)\right). \end{aligned} \tag{87}$$

For $k = \mathrm{HF}$, we vary b and c-jet efficiencies, and for $k = \mathrm{LF}$, we vary the efficiencies for light-quark and gluon jets. Using $\nu_k = \pm 1$, we construct synthetic data sets and fit linear surrogates

$$\hat{S}_{\mathrm{HF}}(\boldsymbol{x}|\nu_{\mathrm{HF}}) = \exp\left(\nu_{\mathrm{HF}} \hat{\Delta}_{\mathrm{HF}}(\boldsymbol{x})\right) \quad \text{and} \quad \hat{S}_{\mathrm{LF}}(\boldsymbol{x}|\nu_{\mathrm{LF}}) = \exp\left(\nu_{\mathrm{LF}} \hat{\Delta}_{\mathrm{LF}}(\boldsymbol{x})\right). \tag{88}$$

The resulting parametrization is shown in figure 8. The HF and LF uncertainties exhibit similar shapes. HF variations range from 2% to 5%, while LF variations show a slightly larger impact, ranging from 4% to 8%. This greater effect of the LF variations is due to the higher light-jet multiplicity following the the $m(\boldsymbol{t\bar{t}}) \geqslant 750$ GeV selection.

7.5. Uncertainties in lepton efficiencies

Uncertainties in lepton efficiencies are detailed in [80–82] and are handled using the weighting function in equation (34). Since the efficiency scale factors and uncertainties depend on the candidate's pseudo-rapidity, which is not included in $\boldsymbol{x}$, we retain the $\boldsymbol{z}$ dependence in

$$r(\boldsymbol{x}_i, \boldsymbol{z}_i|\nu_\ell) = \prod_{\ell=1}^{2} \left(1 + \frac{\Delta_\ell \mathrm{SF}(\ell)}{\mathrm{SF}(\ell)}\right)^{\nu_\ell}, \tag{89}$$

used to define two surrogate data sets corresponding to $\pm 1\sigma$ variations. Here, $\mathcal{V} = \{-1, 1\}$, and we learn a surrogate

$$\hat{S}_\ell(\boldsymbol{x}|\nu_\ell) = \exp\left(\nu_\ell \hat{\Delta}_\ell(\boldsymbol{x})\right). \tag{90}$$

Variations are under 1% in all cases, with minimal $\boldsymbol{x}$-dependence, as shown in figure 9.

7.6. Testing the tree-based estimates with neural networks

The comparisons in previous sections are one-dimensional projections. For a more general check of whether the BPT is fully expressive in the high-dimensional $\boldsymbol{x}$ space, we can apply a 'Classifier two-sample test' (C2ST) [83, 84].

The C2ST is a non-parametric method for assessing if two samples originate from the same distribution. It trains a binary classifier on a combined dataset of the two samples, using labels to indicate sample origin. The classifier's accuracy reveals distribution similarity; accuracy above chance suggests different distributions.

Given a specific nuisance parameter $\boldsymbol{\nu}$ and a pair of synthetic data sets, $\mathcal{D}_{\mathrm{SM}}$ and $\mathcal{D}_\nu$ with $\nu \neq 0$, if a candidate estimate $\hat{S}(\boldsymbol{x}|\boldsymbol{\nu})$ is accurate and fully expressive, then

$$\hat{S}(\boldsymbol{x}|\nu) = \frac{\mathrm{d}\sigma(\boldsymbol{x}|\boldsymbol{\nu})}{\mathrm{d}\sigma(\boldsymbol{x}|\mathrm{SM})}. \tag{91}$$

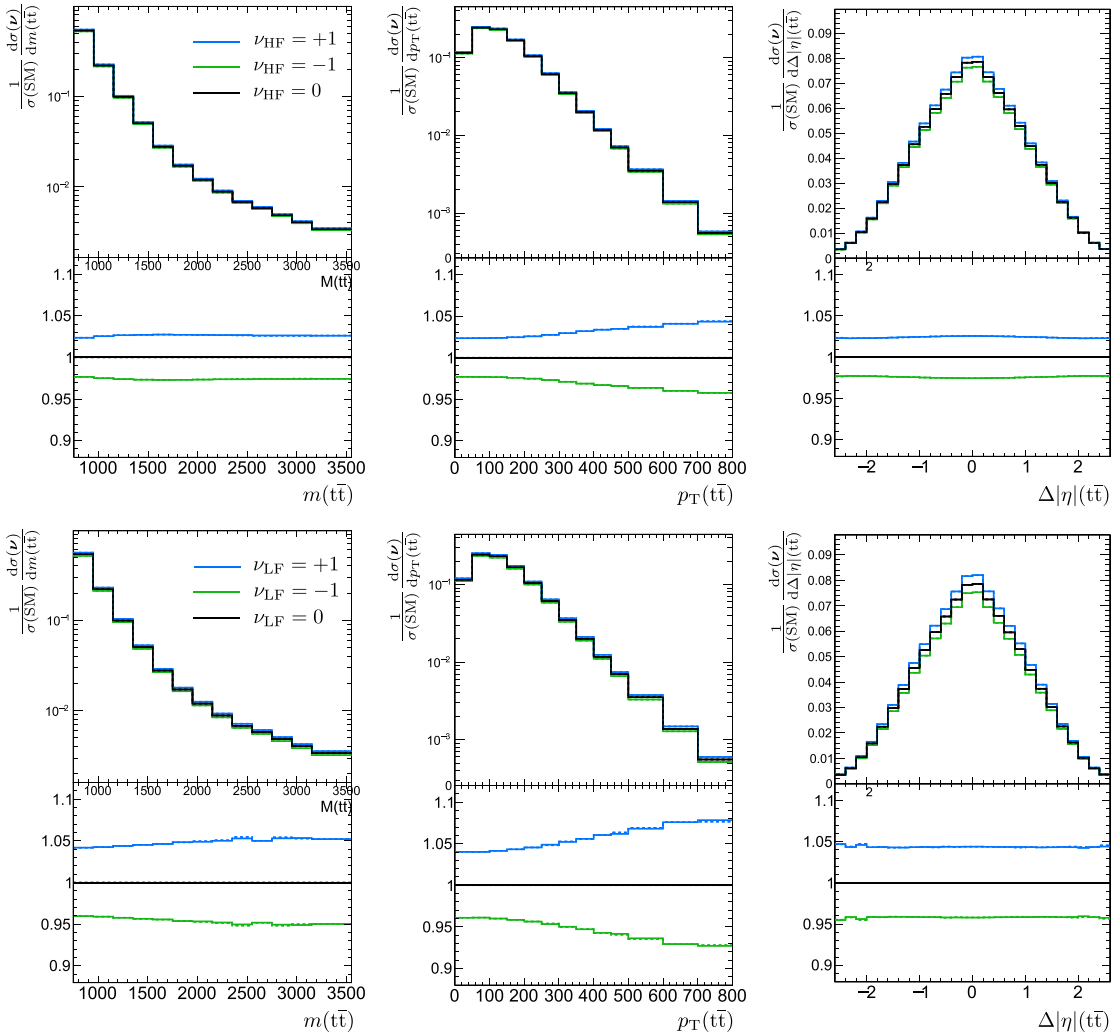

Figure 8. Variations of the nominal simulation with the nuisance parameter for the b-tagging uncertainties $\boldsymbol{\nu}_{\mathrm{HF}}$ and $\boldsymbol{\nu}_{\mathrm{HF}}$. The dashed lines show the training data and the solid lines show the result from the surrogate $\hat{S}_{\mathrm{HF}}(\boldsymbol{x}|\nu_{\mathrm{HF}})$ and $\hat{S}_{\mathrm{LF}}(\boldsymbol{x}|\nu_{\mathrm{LF}})$.

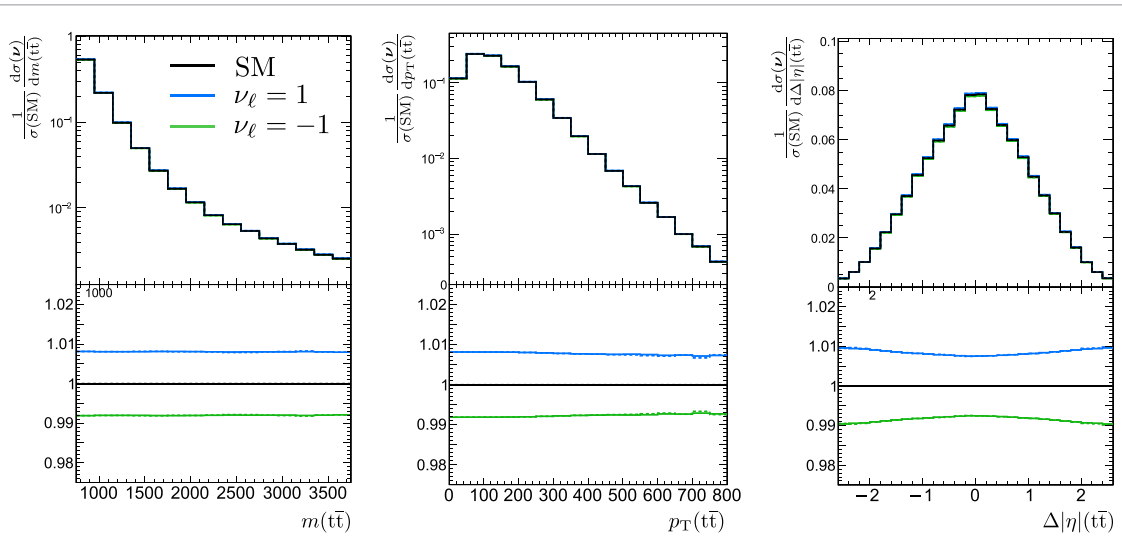

Figure 9. Variations of the nominal simulation with the nuisance parameter for the lepton scale factor uncertainty $\boldsymbol{\nu}_\ell$. The dashed lines show the training data, and the solid lines show the result from the surrogate $\hat{S}_\ell(\boldsymbol{x}|\nu_\ell)$.

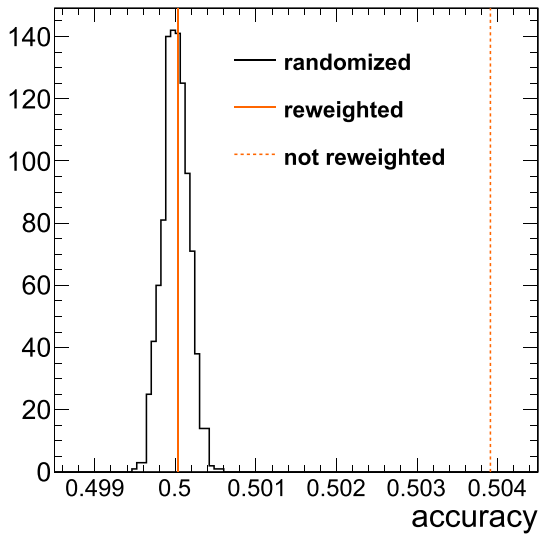

Figure 10. A neural-network classifier tests the surrogate $\hat{S}_{\mathrm{HF}}(\boldsymbol{x}|\nu_{\mathrm{HF}})$ for $\nu_{\mathrm{HF}} = 1$. The black contour shows the distribution of the classifier when the two samples are from the same distribution at $\boldsymbol{\nu}_{\mathrm{HF}} = 0$. The solid orange line indicates the accuracy of a classifier trained with an SM event sample and a sample with $\boldsymbol{\nu}_{\mathrm{HF}} = 1$ that is reweighted back to the SM using the surrogate under test, suggesting no flaw in the performance. The dashed orange line indicates the accuracy of the test trained with two unequal samples at $\boldsymbol{\nu}_{\mathrm{HF}} = 0$ and $\boldsymbol{\nu}_{\mathrm{HF}} = 1$.

For all $\boldsymbol{x}$ and $\boldsymbol{\nu}$. Thus, reweighting $\mathcal{D}_{\boldsymbol{\nu}}$ to form

$$\mathcal{D}_{\mathrm{reweighted}} = \left\{ w_i' = \hat{S}(\boldsymbol{x}|\boldsymbol{\nu})^{-1} w_i, \boldsymbol{x}_i \;\text{ for all }\; w_i, \boldsymbol{x}_i \in \mathcal{D}_{\boldsymbol{\nu}} \right\} \tag{92}$$

Should make $\mathcal{D}_{\mathrm{reweighted}}$ indistinguishable from $\mathcal{D}_{\mathrm{SM}}$. To test this, a classifier's accuracy in distinguishing $\mathcal{D}_{\mathrm{reweighted}}$ from $\mathcal{D}_{\mathrm{SM}}$ is used, with a p-value based on the null distribution of the accuracy. We train a classifier using HF b-tagging with $\nu = \nu_{\mathrm{HF}} = 1$ to test $\hat{S}_{\mathrm{HF}}(\boldsymbol{x}|\nu_{\mathrm{HF}} = 1)$. The classifier, a neural network in `pytorch` with sigmoid activation and three hidden layers (512, 512, 256 units), is optimized with `Adam` on half of the data. Its accuracy is 0.5001, suggesting near-perfect agreement. To evaluate this result, we merge $\mathcal{D}_{\mathrm{reweighted}}$ and $\mathcal{D}_{\mathrm{SM}}$, randomize labels, and train 1000 classifiers on pairs of identical subsets. The null distribution peaks at 0.5, as shown in figure 10. For comparison, distinguishing $\mathcal{D}_{\nu_{\mathrm{HF}}=1}$ from $\mathcal{D}_{\mathrm{SM}}$ yields 0.504, a significant deviation (figure 10). This deviation is small due to the mild $\boldsymbol{x}$-dependence of the DCR, yet the neural network accurately detects the difference. In summary, removing ν_{HF}-dependence with our surrogate makes it impossible for a high-sensitivity neural network to distinguish from the SM, indicating strong performance across the feature space.

7.7. Expected Limits from unbinned Asimov data

The Asimov dataset [85] is commonly used to derive expected exclusion limits from binned Poisson likelihoods [59]. Gomez Ambrosi *et al* [8] extends this to the unbinned case, enabling sampling-free exclusions within continuous parametric models. Here, we consider composite hypotheses involving two Wilson coefficients, which we denote by $\boldsymbol{\theta}$. Under the exclusion scenario, $\boldsymbol{\theta}$ represents the null hypothesis with $N_{\boldsymbol{\theta}} = 2$, while other Wilson coefficients are profiled as nuisance parameters. The alternative hypothesis assumes $\boldsymbol{\theta} = \mathbf{0}$.

Wilks' theorem states that if the data are distributed under the null hypothesis $\boldsymbol{\theta}$, the test statistic $p(q_{\boldsymbol{\theta}}|\boldsymbol{\theta}, \boldsymbol{\nu})$ asymptotically follows a central χ^2 distribution with $N_{\boldsymbol{\theta}}$ degrees of freedom. This distribution is independent of the true values of the nuisance parameters. Given that our POIs primarily influence the predictions linearly [74], we assume any minor quadratic terms do not invalidate Wilks' theorem [27]. However, in practical applications, this assumption should be verified, as shown in [8], where good agreement was observed. Since $q_{\boldsymbol{\theta}}$ is monotonic with the p-value, it can define acceptance regions for $\boldsymbol{\theta}$ at CLs of 68% ($\alpha = 32\%$) or 95% ($\alpha = 5\%$). We anticipate excluding a hypothesis $\boldsymbol{\theta}$ at a given CL if there's a 50% or greater probability for $q_{\boldsymbol{\theta}}$ to fall outside the corresponding acceptance region when the alternate hypothesis $\boldsymbol{\theta} = \mathbf{0}$ is true. Therefore, we must solve

$$\int_{q_{\boldsymbol{\theta},\mathrm{med}}}^{\infty} p(q_{\boldsymbol{\theta}}|\boldsymbol{\theta})\, \mathrm{d}q_{\boldsymbol{\theta}} = \alpha \quad \text{and} \quad q_{\boldsymbol{\theta},\mathrm{med}} = \mathrm{Med}\,(q_{\boldsymbol{\theta}}|\boldsymbol{\theta} = \mathbf{0})\,. \tag{93}$$

The final ingredient is Wald's theorem [86], which implies that the distribution $\mathrm{p}(q_{\boldsymbol{\theta}}|\mathbf{0})$ asymptotically follows a non-central χ^2 distribution with $N_{\boldsymbol{\theta}}$ degrees of freedom and non-centrality parameter Λ. This parameter can be computed (see [8] for details) for the unbinned likelihood ratio. As in the binned case, it corresponds to the Asimov expectation of equation (11) for the alternate hypothesis, multiplied by -2. In our notation, the result is

$$\begin{aligned} -\frac{1}{2}\Lambda &= -\mathcal{L}(\boldsymbol{\nu})\,\sigma(\boldsymbol{\theta},\boldsymbol{\nu}) + \mathcal{L}_0\,\sigma(\mathrm{SM}) + \mathcal{L}_0\,\langle \log(\mathcal{L}(\boldsymbol{\nu})\,\mathcal{R}(\boldsymbol{x}_i|\boldsymbol{\theta},\boldsymbol{\nu})/\mathcal{L}_0)\rangle_{\mathrm{SM}} - \frac{1}{2}\sum_{k=1}^{K}\nu_k^2 \\ &= \sum_{\boldsymbol{x}_i, w_i \in \mathcal{D}_0 \cap \mathcal{X}} w_i\,(-\mathcal{L}(\boldsymbol{\nu})\,\mathcal{R}(\boldsymbol{x}_i|\boldsymbol{\theta},\boldsymbol{\nu}) + \mathcal{L}_0 + \mathcal{L}_0 \log(\mathcal{L}(\boldsymbol{\nu})\,\mathcal{R}(\boldsymbol{x}_i|\boldsymbol{\theta},\boldsymbol{\nu})/\mathcal{L}_0)) - \frac{1}{2}\sum_{k=1}^{K}\nu_k^2. \end{aligned} \tag{94}$$

With the integrated luminosity from equation (81) and the model's DCR $\mathcal{R}(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\nu})$ from equation (82). The sum is over all events in the nominal $\bar{\boldsymbol{t}}\boldsymbol{t}(2\ell)$ sample passing the event selection. The ratio $\mathcal{R}(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\nu})$ appears in both the logarithm and the 'extended' term for the total fiducial cross-section. This expression provides the test statistic under the alternate hypothesis, allowing us to obtain the expected exclusion contour from the profiled likelihood test statistic. The minimization is performed with the IMINUIT package [87].

7.8. Results

Figure 11 shows the Asimov expected exclusion contours at 68% CL (dashed) and 95% CL (solid). The SMEFT effects are simulated up to quadratic order in the POIs. For the blue contours, the other three Wilson coefficients are profiled, while nuisance parameters are set to zero. For the black contours, nuisance parameters are also profiled. Systematic uncertainties significantly impact $\mathrm{Re}[C_{tG}]$, which strongly affects the total yield and is sensitive to integrated luminosity, renormalization and factorization scales, and normalization uncertainties. For non-zero $\mathrm{Im}[C_{tG}]$ and $\mathrm{Re}[C_{tG}] \approx 0$, effects from the three remaining four-fermion operators outweigh those of systematic uncertainties, explaining why the contours degrade only slightly when including systematics. A comprehensive $\bar{\boldsymbol{t}}\boldsymbol{t}(2\ell)$ sensitivity analysis would require analyzing all uncertainties, some of which are not publicly available. This study, however, shows how systematic effects can be captured in machine-learned surrogates and applied in limit setting.

8. Conclusion

This paper presents a comprehensive, scalable framework for modeling the effects of systematic uncertainties in unbinned analyses of collider data. By factorizing systematic effects across parton, particle, and detector levels, we make them accessible for ML. With a highly granular factorization of the various dependencies, we leverage the extensive knowledge gained from binned LHC data analyses and fully capitalize on high-quality MC simulation. A flexible approach facilitates the progressive refinement of unbinned models, including but not restricted to applications in SMEFT. It accommodates new systematic effects or background contributions without invalidating previously trained surrogates.

A significant technical innovation introduced is the BPT, an extension of tree-boosting algorithms designed to learn accurate parametrizations of systematic dependencies. BPTs offer a robust and efficient alternative to neural networks for modeling systematic effects, providing reliable surrogate models for complex, high-dimensional parameter spaces in unbinned hypothesis testing.

Our work thus bridges a critical gap in the methodological toolbox for SMEFT analyses, searches for other non-resonant effects beyond the SM, and similar inference problems. We demonstrate the practical application through a semi-realistic case study of top quark pair production in the dilepton channel, which underscores the effectiveness of our approach in learning and incorporating systematic effects. Overall, the new techniques pave the way for more refined and adaptable unbinned hypothesis tests, enhancing the accuracy and reliability of SMEFT analyses. We anticipate these advancements will be instrumental in exploiting the data from future collider experiments. Finally, we believe that publicly available refined models would be useful for future SMEFT combinations and for providing legacy LHC results.

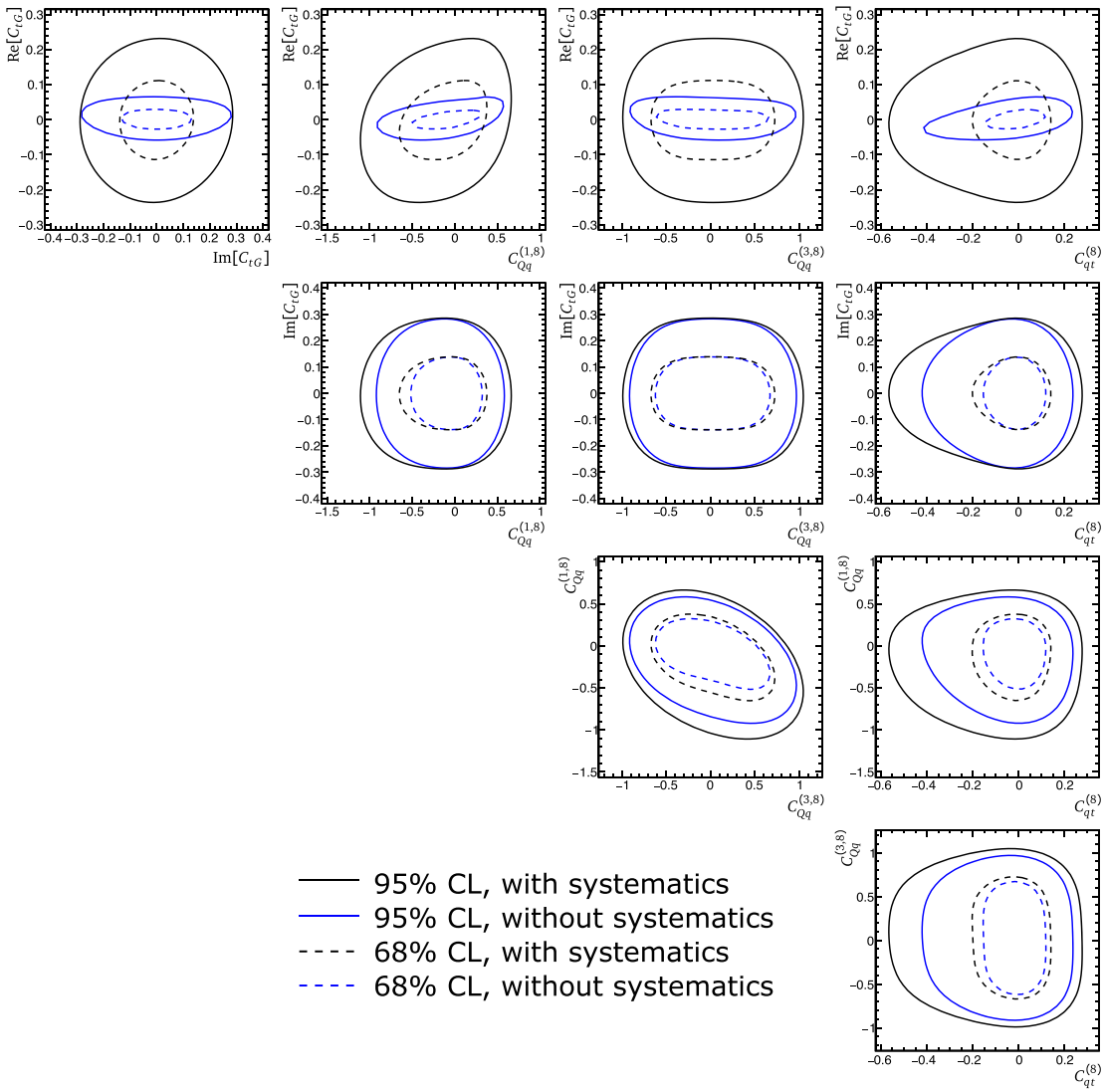

Figure 11. Asimov expected exclusion contours at 68% CL (dashed) and at 95% CL (solid). The three other Wilson coefficients are profiled for the blue contours, and all nuisance parameters are frozen at zero. For the black contours, the three other Wilson coefficients and all the nuisance parameters are profiled.

Data availability statement

No new data were created or analysed in this study.

Acknowledgment

The computational results were obtained using the Vienna Bio Center and the CLIP computing cluster of the Austrian Academy of Sciences at www.clip.science/. I am indebted to Suman Chatterjee, Claudius Krause, Tilman Plehn, Dennis Schwarz, Nick Smith, and Nicholas Wardle for many useful discussions.

Appendix A. Per-event SMEFT weights

We show how to efficiently obtain a polynomial per-event SMEFT parametrization from generator weights obtained at a sufficient number of different values $\boldsymbol{\theta}$ with dimension $N_{\boldsymbol{\theta}}$. The procedure can be extended to arbitrary fixed polynomial order, but for simplicity, we truncate after the quadratic term,

$$w_i(\boldsymbol{\theta}) = \omega_{i,0} + \omega_{i,m}\theta_m + \omega_{i,mn}\theta_m\theta_n. \tag{A.1}$$

We can take the quadratic coefficients for each event as an upper triangular matrix, $\omega_{i,mn} = 0$ for $n < m$ for all events i. From the generator, we can obtain the r.h.s. of equation (A.1) as

$$w_i(\boldsymbol{\theta}) \propto |\mathcal{M}_{\text{SMEFT}}(\boldsymbol{z}_{p,i}|\boldsymbol{\theta})|^2 \tag{A.2}$$

Which we evaluate for $M = 1, \ldots, |M|$ base points. We denote those parameter values by $\boldsymbol{\theta}^M$ and the resulting base point weights by $w_i{}^M = w_i(\boldsymbol{\theta}^M)$. We chose $|M|$ to correspond to the maximum number of independent per-event coefficients so that we have exactly enough base point weights to specify the general polynomial dependence. Therefore,

$$|M| = 1 + N_{\boldsymbol{\theta}} + \frac{1}{2} N_{\boldsymbol{\theta}} (N_{\boldsymbol{\theta}} + 1), \tag{A.3}$$

where the three terms in the sum correspond to the number of independent coefficients corresponding to the constant, the linear, and the quadratic per-event SMEFT dependence. To be explicit, $\theta_{M,m}$ is the value of the mth Wilson coefficient at the Mth base point. For each event i, this gives us the $|M|$ equations

$$w_i{}^M = \omega_{i,0} + \omega_{i,m}\theta_m^M + \omega_{i,mn}\theta_m^M\theta_n^M \tag{A.4}$$

where we use Einstein summation for m and n. This is an $|M| \times |M|$ linear equation in $\omega_{i,0}$, $\omega_{i,m}$, and $\omega_{i,mn}$ with coefficients 1, θ_m^M, and $\theta_m^M\theta_n^M$. Equation (A.4) suggests to relabel the indices $\{(1), (m), (mn)\}$ by a multi-index $\mathrm{K} = 1, \ldots, |M|$ where the 1 represents the constant piece, m the $N_{\boldsymbol{\theta}}$ linear terms, and the ordered pair (mn) the $1/2N_{\boldsymbol{\theta}}(N_{\boldsymbol{\theta}} + 1)$ different quadratic terms. Any value of $\boldsymbol{\theta}$ can then also be represented as an $|M|$-component vector $\theta_K = \{1, \theta_m, \theta_m\theta_n\}_K$ and the $|M|$ base points $\boldsymbol{\theta}^M$ provide the $|M| \times |M|$ matrix

$$C^M{}_K = \left\{1, \theta_m^M, \theta_m^M\theta_n^M\right\}_K. \tag{A.5}$$

Concretely, when $N_{\boldsymbol{\theta}} = 15$, we have 136 values that the indices M and K can take. Equation (A.4) then reads

$$w_i{}^M = C^M{}_K \omega_i{}^K. \tag{A.6}$$

The matrix C and its inverse do not depend on the event as long as the base points are kept when running the generator. The base points must be chosen such that C^{-1} exists. We can now compute the per-event polynomial weight coefficients as

$$\omega_i{}^K = C^{-1K}{}_M w_i{}^M. \tag{A.7}$$

From the per-event base-point weights $w_i{}^M$. With these coefficients, we can now evaluate equation (A.1) for variable $\boldsymbol{\theta}$ as

$$w_i(\boldsymbol{\theta}) = \theta_K C^{-1K}{}_M w_i{}^M. \tag{A.8}$$

Finally, we can break up the index K again, i.e. $K = 1$ will give us the constant coefficient, the $N_{\boldsymbol{\theta}}$ terms $K = m$ will give us the linear event-weight dependence, and the $1/2N_{\boldsymbol{\theta}}(N_{\boldsymbol{\theta}} + 1)$ terms $K = (mn)$ provide the quadratic coefficients. For the SMEFT coefficients in equation (36) we find from equation (A.7)

$$\begin{aligned} r^{(m)}(\boldsymbol{z}_{\mathrm{p},i}) &= \omega_i{}^{(m)}/\omega_{i,0}, \\ r^{(mn)}(\boldsymbol{z}_{\mathrm{p},i}) &= \omega_i{}^{(mn)}/\omega_{i,0}. \end{aligned} \tag{A.9}$$

Appendix B. Alternative loss functions

The solution in equation (42) can be obtained from other loss functions that differ in behavior away from the minimum. An example is the quadratic loss

$$L_{\mathrm{Q}}\left[\hat{f}\right] = \left\langle \hat{f}(\boldsymbol{x})^2 \right\rangle_{\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\theta}_1,\boldsymbol{\nu}_1} + \left\langle \left(1 - \hat{f}(\boldsymbol{x})\right)^2 \right\rangle_{\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\theta}_0,\boldsymbol{\nu}_0} \tag{B.1}$$

Which can be used with synthetic data sets, either with or without reweighting, by following the same steps as in section 5.1. For the latter case, the result is

$$L_{\mathrm{Q}}\left[\hat{f}\right] = \int \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{z}\, p\left(\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\theta}_0,\boldsymbol{\nu}_0\right)\left(r\left(\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\theta}_1,\boldsymbol{\nu}_1,\boldsymbol{\theta}_0,\boldsymbol{\nu}_0\right)\hat{f}(\boldsymbol{x})^2 + \left(1-\hat{f}(\boldsymbol{x})\right)^2\right). \tag{B.2}$$

The same is true for the mean-squared-error loss function

$$L_{\mathrm{MSE}}\left[\hat{f}\right] = \left\langle \left(\hat{f}(\boldsymbol{x}) - r\left(\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\theta}_1,\boldsymbol{\nu}_1,\boldsymbol{\theta}_0,\boldsymbol{\nu}_0\right)\right)^2 \right\rangle_{\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\theta}_0,\boldsymbol{\nu}_0}. \tag{B.3}$$

Its minimum satisfies

$$\frac{\mathrm{d}\sigma\left(\boldsymbol{x}|\boldsymbol{\theta}_1,\boldsymbol{\nu}_1\right)}{\mathrm{d}\sigma\left(\boldsymbol{x}|\boldsymbol{\theta}_0,\boldsymbol{\nu}_0\right)} = f^*_{\mathrm{MSE}}\left(\boldsymbol{x}\right). \tag{B.4}$$

If needed, a version with separate samples is obtained by expanding the square and keeping the $\hat{f}$-dependent terms. The result is

$$L_{\mathrm{MSE}}\left[\hat{f}\right] = \left\langle \hat{f}(\boldsymbol{x})^2 \right\rangle_{\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\theta}_1,\boldsymbol{\nu}_1} - 2\left\langle \hat{f}(\boldsymbol{x}) \right\rangle_{\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\nu}_0,\boldsymbol{\nu}_0}. \tag{B.5}$$

More loss functions can be obtained from the general ansatz

$$L\left[\hat{f}\right] = \left\langle L_1\left[\hat{f}(\boldsymbol{x})\right] \right\rangle_{\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\theta}_1,\boldsymbol{\nu}_1} + \left\langle L_2\left[\hat{f}(\boldsymbol{x})\right] \right\rangle_{\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\theta}_0,\boldsymbol{\nu}_0}, \tag{B.6}$$

where L_1 and L_2, typically, are simple functions of $\hat{f}$. Because it is a sum of expectations over the joint space, this general form allows using the joint-likelihood-ratio in the same way as done for equation (44). Moreover, because $\hat{f}(\boldsymbol{x})$ does not depend on $\boldsymbol{z}$, it is minimized by a function of the ratio of two $\boldsymbol{z}$-integrals (equation (12)) and, therefore, is in one-to-one correspondence with the regression target[6]. If we view the two terms L_1 and L_2 as (standard) functions of $\hat{f}$ and denote the (standard) derivative by L', it is straightforward to show that the conditions

$$-\frac{L_2'}{L_1'} = \frac{1}{\hat{f}} - 1 \qquad \text{and} \qquad -\frac{L_2'}{L_1'} = \hat{f}. \tag{B.7}$$

Lead to loss functions minimized by equation (42) and equation (B.4), respectively. The loss functions discussed so far are special cases of equation (B.7). To control the loss behavior away from the minimum, one can choose, therefore, an appropriate $L_1\left[\hat{f}\right]$ or $L_2\left[\hat{f}\right]$ and compute the other term from equation (B.7).

Appendix C. Construction of the BPT algorithm

This section provides a step-by-step derivation of the BPT algorithm. A summary of the resulting procedures is described in section 5.4.

C.1. Tree-boosting of parametric regressors

It is instructive to discuss boosting for generic non-parametric estimators based on the cross-entropy loss function $L_{\mathrm{CE}}\left[\hat{f}\right]$ in equation (39). After the replacement in equation (47), we have

$$L\left[\hat{T}\right] = \left\langle \mathrm{Soft}^+\left(\hat{T}(\boldsymbol{x})\right) \right\rangle_{\boldsymbol{x},\boldsymbol{z}|\boldsymbol{0}} + \left\langle \mathrm{Soft}^+\left(-\hat{T}(\boldsymbol{x})\right) \right\rangle_{\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\nu}}, \tag{C.1}$$

where we do not yet specify the implementation of $\hat{T}(\boldsymbol{x})$. The loss would attain its minimum at

$$T^*\left(\boldsymbol{x}\right) = \log\frac{\mathrm{d}\sigma\left(\boldsymbol{x}|\boldsymbol{\nu}\right)}{\mathrm{d}\sigma\left(\boldsymbol{x}|\boldsymbol{0}\right)} \tag{C.2}$$

[6] I thank Giuliano Panico for pointing this out.

But instead of obtaining this result in a single fit, we chose a number B of boosting iterations and corresponding learning rates $0 < \eta^{(b)} < 1$ for $b = 1, \ldots, B$. We use an additive expansion of $\hat{T}(\boldsymbol{x})$ in terms of the weak learners $\hat{t}^{(b)}(\boldsymbol{x})$. To this end, we iterate the boosting relations

$$t^{(b)*}(\boldsymbol{x}) = \text{argmin}_{\hat{t}^{(b)}} L\left[\hat{t}^{(b)}(\boldsymbol{x}) + \hat{T}^{(b-1)}(\boldsymbol{x})\right], \tag{C.3}$$

$$\hat{T}^{(b)}(\boldsymbol{x}) = \hat{T}^{(b-1)}(\boldsymbol{x}) + \eta^{(b)}\hat{t}^{(b)*}(\boldsymbol{x}). \tag{C.4}$$

A number of B times, starting with the initial choice $\hat{T}^{(0)}(\boldsymbol{x}) = 0$. Equation (C.3) obtains the weak learner $\hat{t}^{(b)}(\boldsymbol{x})$ when the result of the preceding iteration $\hat{T}^{(b-1)}(\boldsymbol{x})$ is known. Equation (C.4) updates the additive model with a fraction $\eta^{(b)}$ of this weak learner's prediction. After B iterations, the boosted prediction $\hat{T}^{(B)}$ can be expressed as

$$\hat{T}^{(B)}(\boldsymbol{x}) = \sum_{b=1}^{B} \eta^{(b)} t^{(b)*}(\boldsymbol{x}). \tag{C.5}$$

An important practicality for boosting learners that are fit to synthetic data sets follows from the minimum condition in equation (C.3). It implies that the minimum at iteration b satisfies

$$t^{(b)*}(\boldsymbol{x}) + \hat{T}^{(b-1)}(\boldsymbol{x}) \simeq \log \frac{\mathrm{d}\sigma(\boldsymbol{x}|\boldsymbol{\nu})}{\mathrm{d}\sigma(\boldsymbol{x}|\boldsymbol{0})} = \log \frac{\sigma(\boldsymbol{\nu})}{\sigma(\boldsymbol{0})} \frac{\int \mathrm{d}\boldsymbol{z}\, p(\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\nu})}{\int \mathrm{d}\boldsymbol{z}\, p(\boldsymbol{x},\boldsymbol{z}|\boldsymbol{0})}. \tag{C.6}$$

Which we rearrange to

$$t^{(b)*}(\boldsymbol{x}) \simeq \log \frac{\sigma(\boldsymbol{\nu})}{\sigma(\boldsymbol{0})} \frac{\int \mathrm{d}\boldsymbol{z}\, p(\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\nu}) \times \exp\left(-\hat{T}^{(b-1)}(\boldsymbol{x})\right)}{\int \mathrm{d}\boldsymbol{z}\, p(\boldsymbol{x},\boldsymbol{z}|\boldsymbol{0})}. \tag{C.7}$$

By reading this equation as an $\boldsymbol{x}$-dependent scaling of the joint-space integration measure $\mathrm{d}\sigma(\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\nu}) = \sigma(\boldsymbol{\nu})\, p(\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\nu})\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{z}$ by the reciprocal of the estimate of the preceding boosting iteration, we find that $t^{(b)*}$ can also be obtained if, instead of using the additive expansion, we replace $\sigma(\boldsymbol{\nu})\, p(\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\nu}) \to \exp\left(-\hat{T}^{(b-1)}(\boldsymbol{x})\right) \sigma(\boldsymbol{\nu})\, p(\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\nu})$. Because equation (C.4) provides the exponent $\hat{T}$ iteratively, we only have to multiply the cross-section by $\exp(-\eta^{(b-1)} t^{(b-1)*}(\boldsymbol{x}))$ when moving from iteration $b-1$ to iteration b. This way, the boosting equations read

$$\sigma(\boldsymbol{\nu})\, p^{(b)}(\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\nu}) = \exp\left(-\eta^{(b-1)} t^{(b-1)*}(\boldsymbol{x})\right) \sigma(\boldsymbol{\nu})\, p^{(b-1)}(\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\nu}) \tag{C.8}$$

$$t^{(b)*}(\boldsymbol{x}) = \text{argmin}_{\hat{t}^{(b)}} L\left[\hat{t}^{(b)}(\boldsymbol{x})\right], \tag{C.9}$$

$$\hat{T}^{(b)}(\boldsymbol{x}) = \hat{T}^{(b-1)}(\boldsymbol{x}) + \eta^{(b)}\hat{t}^{(b)*}(\boldsymbol{x}) \tag{C.10}$$

Initialized by $\hat{T}^{(0)}(\boldsymbol{x}) = t^{(0)*}(\boldsymbol{x}) = 0$. The advantage of this formulation is that equation (C.9) is a standard loss function minimization without the additive model appearing in the argument as in equation (C.3). The update of the synthetic data set $\mathcal{D}_{\boldsymbol{\nu}}^{(b)} = \{w_i^{(b)}, \boldsymbol{x}_{\boldsymbol{\nu},i}, \boldsymbol{z}_i\}$, now also defined for each iteration b, follows from equation (C.8) as

$$w_i^{(b)} = \exp\left(-\eta^{(b-1)} t^{(b-1)*}(\boldsymbol{x}_i)\right) w_i^{(b-1)}. \tag{C.11}$$

This prescription can be interpreted as a recursive weighting of the differential cross-section of $\mathcal{D}_{\boldsymbol{\nu}}$ in the second term in equation (C.1) towards $\mathcal{D}_{\boldsymbol{0}}$ in the first term. The boosting algorithm removes the learned approximation from the training data as the regressor learns to approximate the DCR more accurately. It is customary to chose $\eta^{(b)}$ independently of b, and values between 10^{-3} and $3 \cdot 10^{-1}$ for this universal learning rate have proven efficient.

The sample $\mathcal{D}_{\boldsymbol{0}}$ stays unchanged in the boosting procedure because we decided to write the $\boldsymbol{x}$-dependent scaling in equation (C.8) in the numerator. The choice of only reweighting the sample $\mathcal{D}_{\boldsymbol{\nu}}$ is a critical detail. It holds the key to a boosting algorithm that works for the fully parametric regressor, including the $\boldsymbol{\nu}$

dependence. We can construct the loss for a parametric tree-based algorithm from the general parametric loss function in equation (51), which is a sum of equally structured terms

$$L = \sum_{\boldsymbol{\nu}\in\mathcal{V}} L_{\text{CE}}\left[\hat{T}(\boldsymbol{x}|\boldsymbol{\nu})\right] = \sum_{\boldsymbol{\nu}\in\mathcal{V}}\left(\left\langle \text{Soft}^{+}\left(\hat{T}(\boldsymbol{x}|\boldsymbol{\nu})\right)\right\rangle_{\boldsymbol{x},z|\mathbf{0}} + \left\langle \text{Soft}^{+}\left(-\hat{T}(\boldsymbol{x}|\boldsymbol{\nu})\right)\right\rangle_{\boldsymbol{x},z|\boldsymbol{\nu}}\right). \tag{C.12}$$

The synthetic data set in the first expectation value in each sum term is always $\mathcal{D}_{\mathbf{0}}$, irrespective of the value of $\boldsymbol{\nu}$. The synthetic data set in the second expectation is $\mathcal{D}_{\boldsymbol{\nu}}$ and is different for each $\boldsymbol{\nu}\in\mathcal{V}$. It is this term whose synthetic data set changes during the boosting algorithm, and because there is one such set for each $\boldsymbol{\nu}\in\mathcal{V}$, the reweighting can be done simultaneously for each $\boldsymbol{\nu}$ in the sum over $\mathcal{V}$ in equation (C.12). Repeating the steps starting at equation (C.1) with a sufficiently expressive $\boldsymbol{\nu}$-dependent function $\hat{T}(\boldsymbol{x}|\boldsymbol{\nu})$, it is straightforward to show that aside from the extra $\boldsymbol{\nu}$-dependence in the notation nothing else changes. Concretely, we only need to modify equation (C.5) to notate the $\boldsymbol{\nu}$-dependence in the weak learner $\hat{t}(\boldsymbol{x}|\boldsymbol{\nu})$. The other steps follow analogously, and equation (C.11) generalizes to

$$\mathcal{D}_{\boldsymbol{\nu}}^{(b)} = \left\{\exp\left(-\eta^{(b-1)} t^{(b-1)*}(\boldsymbol{x}_i|\boldsymbol{\nu})\right) w_i^{(b-1)}, \boldsymbol{x}_i, \boldsymbol{z}_i\right\} \text{ for all } \left\{w^{(b-1)}, \boldsymbol{x}_i, \boldsymbol{z}_i\right\} \in \mathcal{D}_{\boldsymbol{\nu}}^{(b-1)} \text{ for all } \boldsymbol{\nu}\in\mathcal{V} \tag{C.13}$$

Which is the same as equation (C.11) except for that it is carried out simultaneously for each $\boldsymbol{\nu}\in\mathcal{V}$. This completes the boosting algorithm for generic weak learners, and we can proceed with constructing the tree-based implementation.

C.2. Learning the phase-space partitioning

We construct the parametric weak learner $\hat{t}^{(b)}(\boldsymbol{x}|\boldsymbol{\nu})$ in two steps. Because the procedure is identical at each boosting iteration, we drop the superscript (b) in this section in favor of readability and write $\hat{t}(\boldsymbol{x}|\boldsymbol{\nu})$ in place of $\hat{t}^{(b)}(\boldsymbol{x}|\boldsymbol{\nu})$. We first specify the non-linearity in $\boldsymbol{x}$ while keeping a parametric $\boldsymbol{\nu}$-dependence fully general.

We decompose the phase space $\mathcal{X}$ into non-overlapping regions $\Delta\boldsymbol{x}_j$, collectively denoted by $\mathcal{J}$. Such a phase-space partitioning satisfies

$$\mathcal{X} = \bigcup_{j\in\mathcal{J}} \Delta\boldsymbol{x}_j \quad \text{and} \quad \Delta\boldsymbol{x}_j \cap \Delta\boldsymbol{x}_{j'} = \emptyset \iff j \neq j'. \tag{C.14}$$

The nonlinearity in a tree ansatz can always be expressed via the index function

$$\mathbb{1}_j(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \boldsymbol{x}\in\Delta\boldsymbol{x}_j \\ 0 & \text{otherwise.} \end{cases} \tag{C.15}$$

In terms of, for now, arbitrary functions $\hat{t}_j(\boldsymbol{\nu})$ that have no $\boldsymbol{x}$-dependence,

$$\hat{t}(\boldsymbol{x}|\boldsymbol{\nu}) = \sum_{j\in\mathcal{J}} \mathbb{1}_j(\boldsymbol{x})\, \hat{t}_j(\boldsymbol{\nu}). \tag{C.16}$$

The function $\hat{t}_j(\boldsymbol{\nu})$ should describe the DCR in bin j.

Because the $\boldsymbol{x}$-dependence is only in the index function, we can insert equation (C.16) into equation (C.12) and use equation (43) to carry out the event sums over the synthetic data sets. The result is

$$L\left[\mathcal{J}, \hat{t}_j\right] = \sum_{j\in\mathcal{J}} L_j\left[\hat{t}_j\right] = \sum_{j\in\mathcal{J}}\sum_{\boldsymbol{\nu}\in\mathcal{V}}\left[\sigma_{j,0}\,\text{Soft}^{+}\left(\hat{t}_j(\boldsymbol{\nu})\right) + \sigma_{j,\boldsymbol{\nu}}\,\text{Soft}^{+}\left(-\hat{t}_j(\boldsymbol{\nu})\right)\right]. \tag{C.17}$$

The $\sigma_{j,0}$ and $\sigma_{j,\boldsymbol{\nu}}$ are given in terms of the training data as

$$\sigma_{j,0} = \sum_{(\boldsymbol{x}_i,w_i)\in\mathcal{D}_0\cap\Delta\boldsymbol{x}_j} w_i \qquad \text{and} \qquad \sigma_{j,\boldsymbol{\nu}} = \sum_{(\boldsymbol{x}_i,w_i)\in\mathcal{D}_{\boldsymbol{\nu}}\cap\Delta\boldsymbol{x}_j} w_i. \tag{C.18}$$

And can be understood as the synthetic predictions for the cross-section in bin $j\in\mathcal{J}$ for nuisance parameters $\mathbf{0}$ and $\boldsymbol{\nu}$, respectively. We have now decomposed our problem into two related problems that each pertain to different trainable parameters: the phase space partitioning $\mathcal{J}$ and, independently in each region of the partitioning, a function $\hat{t}_j(\boldsymbol{\nu})$ whose $\boldsymbol{\nu}$-dependence we still have to specify.

Before we tackle these problems, it is instructive to develop an intuition for the loss function in equation (C.17). We assume an infinitely expressive $\hat{t}_j(\boldsymbol{\nu})$ and functionally differentiate Equation (C.17) to arrive at

$$0 = \frac{\delta L_j}{\delta \hat{t}_j} = \sum_{\boldsymbol{\nu}\in\mathcal{V}} \left[\frac{\sigma_{j,\mathbf{0}}}{1+\exp\left(-\hat{t}_j(\boldsymbol{\nu})\right)} - \frac{\sigma_{j,\boldsymbol{\nu}}}{1+\exp\left(\hat{t}_j(\boldsymbol{\nu})\right)} \right]. \tag{C.19}$$

This equation is satisfied exactly if

$$\hat{t}_j(\boldsymbol{\nu}) = \log \frac{\sigma_{j,\boldsymbol{\nu}}}{\sigma_{j,\mathbf{0}}} \quad \text{for all} \quad \boldsymbol{\nu} \in \mathcal{V}. \tag{C.20}$$

Which we can always fulfill for sufficiently expressive $\hat{t}_j(\boldsymbol{\nu})$; a perfect representation of the $\boldsymbol{\nu}$-dependence in each tree node $j \in \mathcal{J}$ reproduces the logarithm of the DCR for those values of $\boldsymbol{\nu}$ whose synthetic data sets we included in the loss function. The predictive function of a tree is finitely expressive. Hence, we will seek an approximation for equation (C.20) in the next section. But we can meanwhile use the result to shed light on the loss function equation (C.17). Formally eliminating $\hat{t}_j(\boldsymbol{\nu})$ in favor of its predictions at the points $\boldsymbol{\nu} \in \mathcal{V}$, we express the loss solely in terms of the per-bin cross-sections $\sigma_{j,\mathbf{0}}$ and $\sigma_{j,\boldsymbol{\nu}}$,

$$L[\mathcal{J}] = \sum_{j\in\mathcal{J}} \sum_{\boldsymbol{\nu}\in\mathcal{V}} \left[\sigma_{j,\mathbf{0}} \log\left(1+\frac{\sigma_{j,\boldsymbol{\nu}}}{\sigma_{j,\mathbf{0}}}\right) + \sigma_{j,\boldsymbol{\nu}} \log\left(1+\frac{\sigma_{j,\mathbf{0}}}{\sigma_{j,\boldsymbol{\nu}}}\right) \right]. \tag{C.21}$$

This equation would provide a loss function for finding the optimal phase space partitioning if we did not need to use finitely expressive $\hat{t}_j(\boldsymbol{\nu})$. If we assume small $\boldsymbol{\nu}$ such that a Taylor expansion of $\sigma_{j,\boldsymbol{\nu}}$ around $\boldsymbol{\nu} = \mathbf{0}$ is a good approximation, we get

$$L[\mathcal{J}] = -\frac{1}{4} \sum_{j\in\mathcal{J}} \sum_{\boldsymbol{\nu}\in\mathcal{V}} \nu_a \nu_b I_{(ab),j} + \dots \tag{C.22}$$

where the ellipsis comprise $\mathcal{O}(\nu^3)$ terms and $\mathcal{J}$-independent contributions. The quantity

$$I_{(ab),j} = \frac{1}{\sigma_{j,\mathbf{0}}} \frac{\partial \sigma_{j,\boldsymbol{\nu}}}{\partial \nu_a} \frac{\partial \sigma_{j,\boldsymbol{\nu}}}{\partial \nu_a} \bigg|_{\boldsymbol{\nu}=\mathbf{0}} \tag{C.23}$$

is the leading contribution in $L[\mathcal{J}]$ and represents the Fisher information matrix of a Poisson measurement in bin j regarding the model parameters. We thus show that our loss function will guide the algorithm towards finding a partitioning $\mathcal{J}$ that maximizes the sum of the Fisher information over all terminal tree nodes.

C.3. Terminal node predictions

The second and final constructive step is to curtail the $\boldsymbol{\nu}$-dependence of $\hat{t}_j$ for each node in the weak learner. We can choose it in analogy to the binned case as it will turn out. We use the ansatz

$$\hat{t}_j(\boldsymbol{\nu}) = \nu_a \hat{\Delta}_{a,j} + \nu_a \nu_b \hat{\Delta}_{ab,j} + \nu_a \nu_b \nu_c \hat{\Delta}_{abc,j} + \dots = \nu_A \hat{\Delta}_{A,j} \tag{C.24}$$

with the multi-index notation as explained in section 5.2. The polynomial order and the coefficients at each polynomial order are truncated to the application's required accuracy. We also allow for its fine-tuning by excluding some of the terms in the polynomial for application-specific reasons. If the node j is small enough that the DCR does not significantly vary with $\boldsymbol{x}$, we get for the first term

$$\hat{\Delta}_{j,a} \approx \frac{\partial}{\partial \nu_a} t_j(\boldsymbol{\nu}) \bigg|_{\boldsymbol{\nu}=\mathbf{0}} = \frac{\partial}{\partial \nu_a} \log \frac{\mathrm{d}\sigma(\boldsymbol{x}|\boldsymbol{\nu})}{\mathrm{d}\sigma(\boldsymbol{x}|\mathbf{0})} \bigg|_{\boldsymbol{\nu}=\mathbf{0}} = s_a + \frac{\partial}{\partial \nu_a} \log \sigma(\boldsymbol{\nu}) \bigg|_{\boldsymbol{\nu}=\mathbf{0}} \quad \text{for} \quad \boldsymbol{x} \in \Delta\boldsymbol{x}_j. \tag{C.25}$$

The last expression relates $\hat{\Delta}_{j,a}$ to the well-known score vector s_a, a sufficient statistic for small $\boldsymbol{\nu}$ and, therefore, an optimal observable. The log-derivative of the inclusive cross-section in the last term does not depend on the phase-space partitioning and, thus, is irrelevant to the optimization. The algorithm will, therefore, aim to reduce the expectation of the variance of the score in the training sample. This is, by definition, the negative value of the Fisher information matrix, consistent with the interpretation in the

preceding section. Depending on the concrete problem and the desired accuracy, the higher-order terms in equation (C.25) can improve the parametrization for larger values of $\boldsymbol{\nu}$.

We now determine the terminal node predictions of a weak learner up to a fixed arbitrary polynomial ordering $\boldsymbol{\nu}$ by computing $\hat{\Delta}_{A,j}$. The parametric tree ansatz is

$$\hat{t}(\boldsymbol{x}|\boldsymbol{\nu}) = \sum_{j\in\mathcal{J}} \mathbb{1}_j(\boldsymbol{x}) \left(\nu_A \hat{\Delta}_{A,j}\right). \tag{C.26}$$

An exact solution cannot be obtained for the optimal values of $\hat{\Delta}$ in the general case because the resulting equations

$$\nu_A \hat{\Delta}_{A,j} = \log \frac{\sigma_{j,\boldsymbol{\nu}}}{\sigma_{j,\mathbf{0}}} \quad \text{for all} \quad \boldsymbol{\nu} \in \mathcal{V} \tag{C.27}$$

are overdetermined if $|\mathcal{V}| > N_\Delta$. We note that equation (C.27) has the same form as equation (C.20) except for the finite expressivity on the l.h.s. We are content with an approximate solution of the per-node parametrization because boosting the weak learner will iteratively reduce the shortcomings either way. Any deficiency of a concrete weak learner will be reduced in the subsequent boosting iteration. For $|\mathcal{V}| < N_\Delta$, the training data cannot provide a unique estimate, and more data sets must be obtained.

The simplest approach for approximately solving equation (C.27) is by minimizing the mean-squared error separately for each node $j \in \mathcal{J}$,

$$L_{j,\mathrm{MSE}}\left[\hat{\Delta}\right] = \sum_{\boldsymbol{\nu}\in\mathcal{V}} \left(\nu_A \hat{\Delta}_{A,j} - \log \frac{\sigma_{j,\boldsymbol{\nu}}}{\sigma_{j,\mathbf{0}}}\right)^2. \tag{C.28}$$

It is solved by

$$\hat{\Delta}_{A,j} = \left[\sum_{\nu\in\mathcal{V}} \boldsymbol{\nu}\boldsymbol{\nu}^{\mathrm{T}}\right]^{-1}_{AB} \left[\sum_{\boldsymbol{\nu}\in\mathcal{V}} \boldsymbol{\nu} \log \frac{\sigma_{j,\boldsymbol{\nu}}}{\sigma_{j,\mathbf{0}}}\right]_B. \tag{C.29}$$

The matrix

$$V_{AB} = \left[\sum_{\nu\in\mathcal{V}} \boldsymbol{\nu}\boldsymbol{\nu}^{\mathrm{T}}\right]_{AB}, \tag{C.30}$$

appearing in the approximate solution, is invertible if the base point coordinate matrix has full rank, as we have assumed in section 5.2.

It is instructive to check that the weak learner appropriately responds to training data that is perfectly, not just approximately, consistent with the polynomial ansatz. If we take constants δ_A and consider a model that predicts $\sigma_{j,\boldsymbol{\nu}} = \exp(\nu_A \delta_A)\sigma_{j,\mathbf{0}}$ in a given region, we can insert into equation (C.29) and find $\hat{\Delta}_A = \delta_A$, confirming that the algorithm learns the exact solution if it has the chance.

To complete the construction of the weak learner, we insert the ansatz equation (C.26) into equation (C.17) and get

$$L[\mathcal{J}] = \sum_{j\in\mathcal{J}} \sum_{\boldsymbol{\nu}\in\mathcal{V}} \left[\sigma_{j,\mathbf{0}}\,\mathrm{Soft}^+\left(\nu_A \hat{\Delta}_{j,A}\right) + \sigma_{j,\boldsymbol{\nu}}\,\mathrm{Soft}^+\left(-\nu_A \hat{\Delta}_{j,A}\right)\right], \tag{C.31}$$

where $\hat{\Delta}_{A,j}$ are obtained from equation (C.29) and $\sigma_{j,0}$ and $\sigma_{j,\boldsymbol{\nu}}$ from equation (C.18). The data samples $\mathcal{D}_\mathbf{0}$, used for the prediction of $\sigma_{j,\mathbf{0}}$ in the first term, can either taken to be the same or statistically independent samples. This loss function is amenable to standard tree algorithms, for example, the CART algorithm or the TAO [55–58] algorithm, both providing tree structures with a hierarchical selection using the features $\boldsymbol{x}$ and that satisfy the requirements in equation (C.14). These algorithms proceed by recursively splitting the training data along either axis-aligned (for CART) or linear combinations of the input features (for TAO), reducing the loss at each iteration. The maximum iteration depth and the minimum number of events in each node are hyperparameters that regularize the fit. If no more splits can be performed, the terminal selections (nodes) represent a phase space partitioning $\mathcal{J}$ of the form in equation (C.14) and the quantities $\hat{\Delta}_j$ can be computed from equation (C.29). The tree then estimates the (log-)DCR as in equation (C.26). As a

function of $\boldsymbol{x}$, the prediction of a single tree changes discontinuously if $\boldsymbol{x}$ traverses a boundary between nodes, and the possibly poor approximation close to the boundaries weakens the learner. Utilizing boosting, i.e. using a sequence of trees in equation (C.3) and equation (C.4), we recover the smooth behavior in $\boldsymbol{x}$ of an arbitrarily expressive regressor. Because each tree in the boosted result in equation (C.5) is parametric in $\boldsymbol{\nu}$, so is the final parametric regression tree.

C.4. Algorithm summary

We can combine the steps in sections C.1–C.3 to summarize the BPT algorithm. It is an iterative fit of a tree-based weak learner with the loss in equation (C.31) to the residuals of the preceding boosting iteration whose predictions are obtained from equations (C.8–C.11). Concretely, we start with training data $\mathcal{D}_0$ at a reference point and several synthetic data sets associated with model parameters $\boldsymbol{\nu} \in \mathcal{V}$. We must have enough data so V_{AB} has full rank. We fit a weak learner using the CART algorithm. At each iteration, the CART algorithm recursively divides the feature space by greedily selecting the dimension and cut value combination that minimizes the loss function. Overfitting is mitigated by enforcing a maximum tree depth and a minimum number of events in each terminal node. We construct new synthetic data from the weak learner's prediction using equation (C.11) to replace $\mathcal{D}_{\boldsymbol{\nu}}$. This reweighting procedure brings the samples $\mathcal{D}_{\boldsymbol{\nu}}$ closer to $\mathcal{D}_0$ by an amount controlled by the learning rate η. We iterate the whole procedure B times and obtain the final result from equation (C.5) as

$$\hat{T}(\boldsymbol{x}|\boldsymbol{\nu}) = \log \hat{S}(\boldsymbol{x}|\boldsymbol{\nu}) = \sum_{b=1}^{B} \eta^{(b)} \sum_{j \in \mathcal{J}^{(b)}} \mathbb{1}_j(\boldsymbol{x})\, \nu_A \hat{\Delta}^{(b)}_{A,j}, \tag{C.32}$$

where $\boldsymbol{\nu}$ is the model parameter we like to predict for and $\mathcal{J}^{(b)}$ is the phase-space partitioning obtained from the CART algorithm. The $\hat{\Delta}^{(b)}_{A,j}$ are the polynomial coefficients of the DCR parametrization in the terminal node j at boosting iteration b. Algorithm 1 is a pseudo-code summary of these steps and defines the parametric regression tree algorithm. It is efficiently implemented using the `Numpy` package [88] and available at [89].

C.5. An analytic toy example

To illustrate the BPT, we consider an arbitrarily chosen one-dimensional two-parameter model

$$\mathrm{d}\sigma(\boldsymbol{x}|\nu_1,\nu_2) = N \exp\left(0.25\,(\nu_1 \sin(x) + \nu_2 \cos(0.5x))^2\right) \mathrm{d}x \tag{C.33}$$

with support $x \in [-\pi, \pi]$. The logarithm of the cross-section is a quadratic polynomial in $\boldsymbol{\nu}$ for all $\boldsymbol{x}$, suggesting a perfect fit with a two-parameter parametric tree at quadratic accuracy. For the training, we chose five base points $\mathcal{V} = \{(0.5,0), (0,0.5), (1,0), (0,1), (0.5,0.5)\}$ that lead to a full-rank matrix V in equation (C.30). With a nominal data set at $(\nu_1,\nu_2) = (0,0)$, we have six synthetic data sets, each with $5 \cdot 10^5$ events, sufficient to train the algorithm. We fit $B = 100$ boosting iterations and require a maximum tree depth of 4 and a minimum requirement for the number of events in each terminal node, which is 50 events. The learning rate is set to 0.2 for all boosting iterations.

In figure 12, we compare the true and predicted values for the DCR for various model parameters. The model parameter configurations include the training and new synthetic data, which are absent during training. After only five iterations, the prediction begins to resemble the true DCRs. After 100 iterations, the fit is nearly perfect; dashed lines show the true DCR from the training data and are not separately visible because of the fit's quality, including the parameter configurations not used during training. In realistic applications, the logarithms of the true DCRs will not be exactly polynomial, mandating some degree of validation of the fit quality on unseen data.

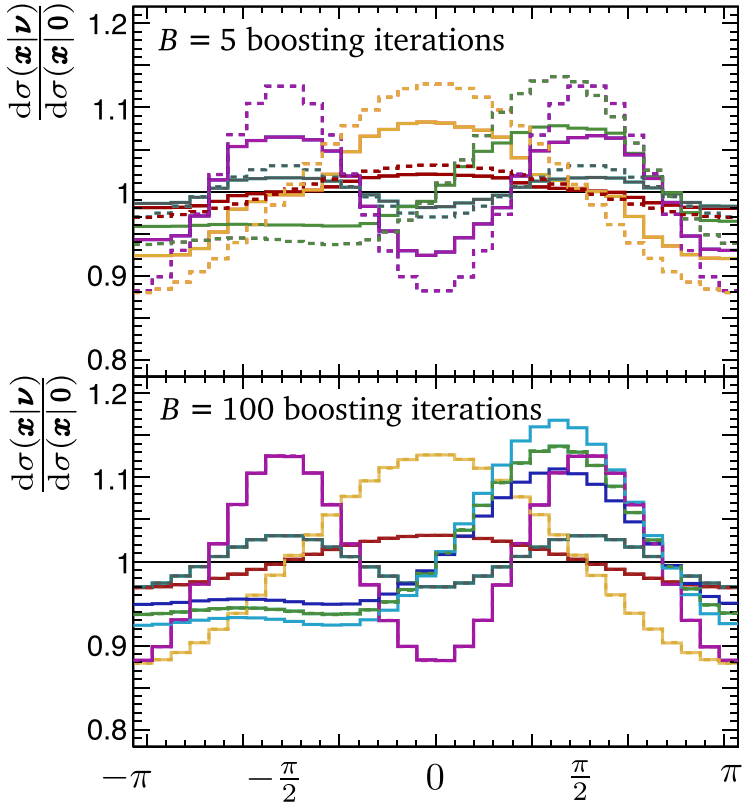

Figure 12. The DCR for various model parameters for the toy study described in the text. Dashed lines show the true DCR from the training data; solid lines show the prediction obtained from the BPT algorithm. The top (bottom) panel compares after $B = 5$ ($B = 100$) boosting iterations. The bottom panel also includes new model parameter points not present during training. In this panel, the prediction is drawn after the truth, and the dashed histograms are not separately visible because of the good quality of the fit with $B = 100$.

Appendix D. Additional angular observables in the $t\bar{t}(2\ell)$ final state

We briefly describe the angular observables introduced in [78]. A measurement of these quantities is performed in [77]. After reconstructing the top-quark momenta, the event is boosted into the $t\bar{t}$ rest frame, and the following axes are defined. The axis $\hat{k}$ points toward the positively charged top quark. The axis $\hat{r}$ is orthogonal to $\hat{k}$ and must lie within the beam plane, spanned by the $\hat{k}$ and the momentum of the incoming parton in the $t\bar{t}$ rest frame. The axis $\hat{n}$ is orthogonal to the beam plane, and $\{\hat{r}, \hat{k}, \hat{n}\}$ must form a right-handed orthonormal basis. The lepton directions of flight, denoted by $\hat{\ell}^+$ and $\hat{\ell}^-$, are measured in the corresponding top quark center-of-mass frame, which is reached from the $t\bar{t}$ frame by a rotation-free Lorentz transformation. Then, the quantities $\xi_{ab} = \cos\theta_a^+ \cos\theta_b^-$ are defined where $\cos\theta_a^+ = \hat{\ell}^+ \cdot \hat{a}$ and $\cos\theta_b^- = \hat{\ell}^- \cdot \hat{a}$ and the axis a and b can each be one of $\{\hat{r}, \hat{k}, \hat{n}\}$. For $a \neq b$, sums and differences of these are considered, e.g. $\xi_{nr}^{\pm} = \xi_{nr} \pm \xi_{rn}$ and analogously for the other combinations. Two more axes $\hat{r}^*$ and $\hat{k}^*$ are defined by flipping the direction of $\hat{r}$ and $\hat{k}$ depending on the sign of the top quarks' rapidity difference in the laboratory frame while keeping the system orthonormal. The resulting 12 independent quantities characterize the spin-density matrix of the $t\bar{t}(2\ell)$ system. More details, including the behavior of these quantities under the discrete SM symmetries, are provided in [78]. We show the distribution of the 12 quantities in figure 13.

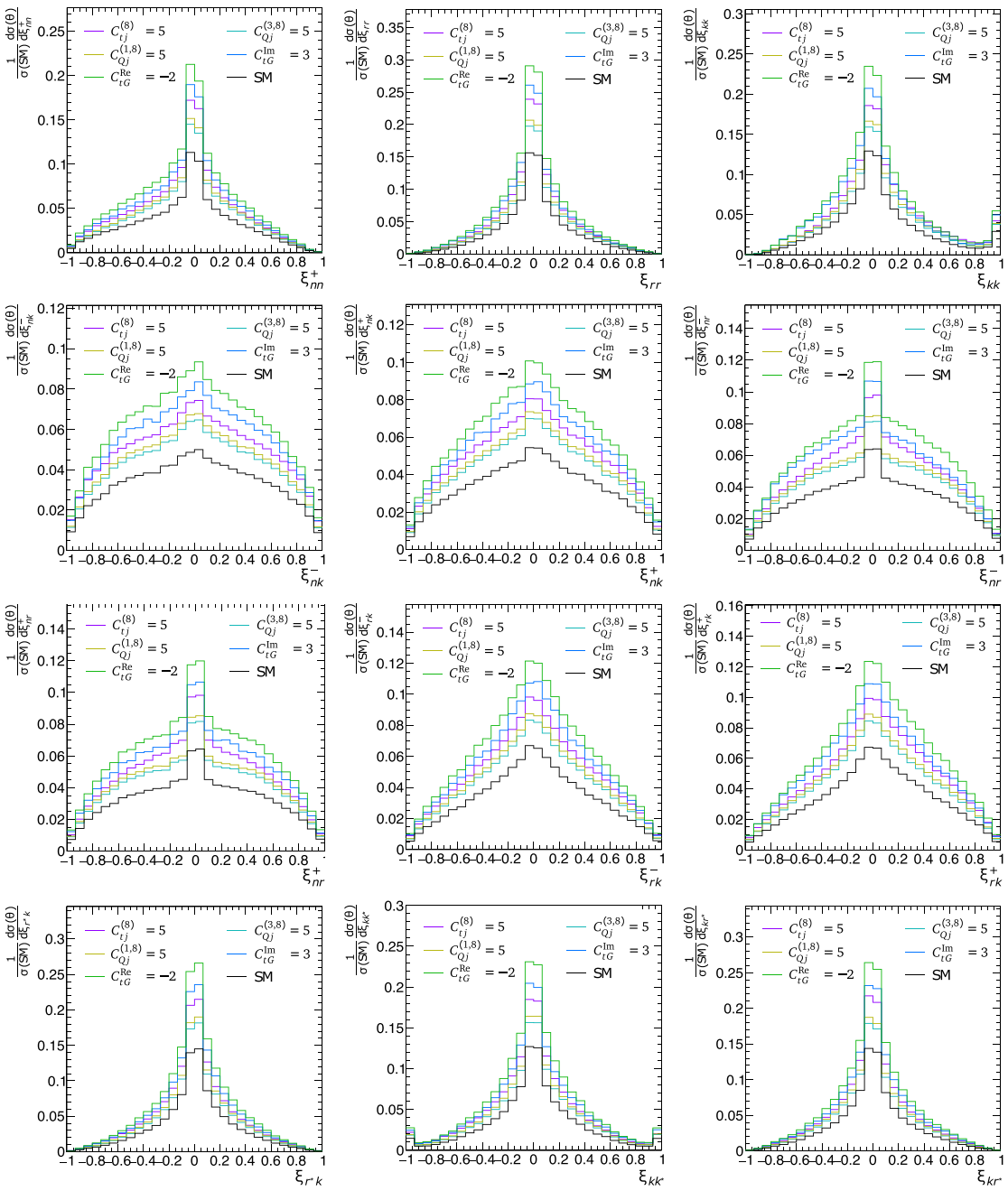

Figure 13. Same as figure 4 for the distribution of the products of leptonic observables described in the text.

ORCID iD

Robert Schöfbeck https://orcid.org/0000-0002-2332-8784

References

[1] Buchmuller W and Wyler D 1986 Effective Lagrangian analysis of new interactions and flavor conservation *Nucl. Phys.* B **268** 621
[2] Leung C N, Love S T and Rao S 1986 Low-energy manifestations of a new interaction scale: operator analysis *Z. Phys.* C **31** 433
[3] Degrande C *et al* 2013 Effective field theory: a modern approach to anomalous couplings *Ann. Phys.* **335** 21
[4] Brivio I and Trott M 2019 The standard model as an effective field theory *Phys. Rep.* **793** 1
[5] Isidori G, Wilsch F and Wyler D 2024 The standard model effective field theory at work *Rev. Mod. Phys.* **96** 015006
[6] Grzadkowski B, Iskrzynski M, Misiak M and Rosiek J 2010 Dimension-six terms in the standard model Lagrangian *J. High Energy Phys.* JHEP10(2010)085
[7] Belvedere A *et al* 2024 LHC EFT WG Note: SMEFT predictions, event reweighting, and simulation *SciPost Physics Community Reports* 4
[8] Gomez Ambrosio R *et al* 2023 Unbinned multivariate observables for global SMEFT analyses from machine learning *J. High Energy Phys.* JHEP03(2023)033

[9] Chatterjee S, Rohshap S, Schöfbeck R and Schwarz D 2022 Learning the EFT likelihood with tree boosting (arXiv:2205.12976)
[10] Chatterjee S *et al* 2022 Tree boosting for learning EFT parameters *Comput. Phys. Commun.* **277** 108385
[11] Chen S, Glioti A, Panico G and Wulzer A 2021 Parametrized classifiers for optimal EFT sensitivity *J. High Energy Phys.* JHEP05(2021)247
[12] Chen S, Glioti A, Panico G and Wulzer A 2024 Boosting likelihood learning with event reweighting *J. High Energy Phys.* JHEP03(2024)117
[13] Cranmer K, Pavez J and Louppe G 2015 Approximating likelihood ratios with calibrated discriminative classifiers (arXiv:1506.02169)
[14] Brehmer J, Cranmer K, Louppe G and Pavez J 2018 Constraining effective field theories with machine learning *Phys. Rev. Lett.* **121** 111801
[15] Brehmer J, Cranmer K, Louppe G and Pavez J 2018 A guide to constraining effective field theories with machine learning *Phys. Rev.* D **98** 052004
[16] Brehmer J, Louppe G, Pavez J and Cranmer K 2020 Mining gold from implicit models to improve likelihood-free inference *Proc. Natl Acad. Sci.* **117** 5242
[17] Brehmer J, Kling F, Espejo I and Cranmer K 2020 MadMiner: machine learning-based inference for particle physics *Comput. Softw. Big Sci.* **4** 3
[18] Brehmer J *et al* 2019 Benchmarking simplified template cross sections in WH production *J. High Energy Phys.* JHEP11(2019)034
[19] Butter A, Plehn T, Soybelman N and Brehmer J 2021 Back to the formula – LHC edn (arXiv:2109.10414)
[20] Plehn T *et al* 2022 Modern machine learning for LHC physicists (arXiv:2211.01421)
[21] Cranmer K 2014 Practical Statistics for the LHC *2011 European School of High-Energy Physics* p 267
[22] d'Agnolo R T *et al* 2022 Learning new physics from an imperfect machine *Eur. Phys. J.* C **82** 275
[23] De Castro P and Dorigo T 2019 INFERNO: inference-aware neural optimisation *Comput. Phys. Commun.* **244** 170
[24] Layer L, Dorigo T and Strong G 2023 Application of inferno to a top pair cross section measurement with CMS open data (arXiv:2301.10358)
[25] Neyman J and Pearson E S 1933 On the problem of the most efficient tests of statistical hypotheses *Phil. Trans. R. Soc.* A **231** 289
[26] Wilks S S 1938 The large-sample distribution of the likelihood ratio for testing composite hypotheses *Ann. Math. Stat.* **9** 60
[27] Bernlochner F U, Fry D C, Menary S B and Persson E 2023 Cover your bases: asymptotic distributions of the profile likelihood ratio when constraining effective field theories in high-energy physics *SciPost Phys. Core* **6** 013
[28] CMS Collaboration 2021 Precision luminosity measurement in proton-proton collisions at $\sqrt{s} = 13$ TeV in 2015 and 2016 at CMS *Eur. Phys. J.* C **81** 800
[29] ATLAS Collaboration 2023 Luminosity determination in *pp* collisions at $\sqrt{s} = 13$ TeV using the ATLAS detector at the LHC *Eur. Phys. J.* C **83** 982
[30] Campbell J M *et al* 2024 Event generators for high-energy physics experiments *SciPost Phys.* **16** 130
[31] Alwall J *et al* 2014 The automated computation of tree-level and next-to-leading order differential cross sections and their matching to parton shower simulations *J. High Energy Phys.* JHEP07(2014)079
[32] Frederix R and Frixione S 2012 Merging meets matching in MC@NLO *J. High Energy Phys.* JHEP12(2012)061
[33] Sherpa Collaboration 2019 Event generation with sherpa 2.2 *SciPost Phys.* **7** 034
[34] Nason P 2004 A New method for combining NLO QCD with shower Monte Carlo algorithms *J. High Energy Phys.* JHEP11(2004)040
[35] Frixione S, Nason P and Oleari C 2007 Matching NLO QCD computations with Parton Shower simulations: the POWHEG method *J. High Energy Phys.* JHEP11(2007)070
[36] Frixione S, Nason P and Ridolfi G 2007 A Positive-weight next-to-leading-order Monte Carlo for heavy flavour hadroproduction *J. High Energy Phys.* JHEP09(2007)126
[37] Alioli S, Nason P, Oleari C and Re E 2010 A general framework for implementing NLO calculations in shower Monte Carlo programs: the POWHEG BOX *J. High Energy Phys.* JHEP06(2010)043
[38] Campbell J M, Ellis R K, Nason P and Re E 2015 Top-pair production and decay at NLO matched with parton showers *J. High Energy Phys.* JHEP04(2015)114
[39] Sjöstrand T *et al* 2015 An introduction to PYTHIA 8.2 *Comput. Phys. Commun.* **191** 159
[40] Bellm J *et al* 2016 Herwig 7.0/Herwig++ 3.0 release note *Eur. Phys. J.* C **76** 196
[41] GEANT4 Collaboration 2003 GEANT4–a simulation toolkit *Nucl. Instrum. Methods* A **506** 250
[42] ATLAS Collaboration 2017 Jet reconstruction and performance using particle flow with the ATLAS Detector *Eur. Phys. J.* C **77** 466
[43] CMS Collaboration 2017 Particle-flow reconstruction and global event description with the CMS detector *JINST* **12** 10003
[44] DELPHES 3 Collaboration 2014 DELPHES 3, a modular framework for fast simulation of a generic collider experiment *J. High Energy Phys.* JHEP02(2014)057
[45] Komiske P T, Metodiev E M and Thaler J 2019 Energy flow networks: deep sets for particle jets *J. High Energy Phys.* JHEP01(2019)121
[46] Chatterjee S, Cruz S S, Schöfbeck R and Schwarz D 2024 Rotation-equivariant graph neural network for learning hadronic SMEFT effects *Phys. Rev.* D **109** 076012
[47] Buckley A *et al* 2015 LHAPDF6: parton density access in the LHC precision era *Eur. Phys. J.* C **75** 132
[48] Mattelaer O 2016 On the maximal use of Monte Carlo samples: re-weighting events at NLO accuracy *Eur. Phys. J.* C **76** 674
[49] Serkin L ATLAS and CMS Collaboration 2021 Treatment of top-quark backgrounds in extreme phase spaces: the "top p_T reweighting" and novel data-driven estimations in ATLAS and CMS *13th Int. Workshop on Top Quark Physics* p 5
[50] Frixione S and Webber B R 2002 Matching NLO QCD computations and parton shower simulations *J. High Energy Phys.* JHEP06(2002)029
[51] Hoeche S *et al* 2005 Matching parton showers and matrix elements *HERA and the LHC: A Workshop on the Implications of HERA for LHC Physics: CERN - DESY Workshop 2004/2005 Midterm Meeting, CERN, (11 October–13 October 2004) Final Meeting, DESY, 17 January–21 January 2005* p 288
[52] Finke T *et al* 2024 Tree-based algorithms for weakly supervised anomaly detection *Phys. Rev.* D **109** 034033
[53] Speckmayer P, Hocker A, Stelzer J and Voss H 2010 The toolkit for multivariate data analysis, TMVA 4 *J. Phys. Conf.: Ser.* **219** 032057
[54] Breiman L, Friedman J H, Olshen R A and Stone C J 1984 *Classification and Regression Trees* (Wadsworth Publishing Company)
[55] Zharmagambetov A, Hada S S, Gabidolla M and Carreira-Perpiñán M A 2021 Non-greedy algorithms for decision tree optimization: An experimental comparison *2021 Int. Joint Conf. on Neural Networks (IJCNN)* p 1

[56] Gabidolla M and Carreira-Perpiñán M A 2022 Pushing the envelope of gradient boosting forests via globally-optimized oblique trees *2022 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)* p 285
[57] Zharmagambetov A and Carreira-Perpiñán M A 2020 Smaller, more accurate regression forests using tree alternating optimization (available at: https://proceedings.mlr.press/v119/zharmagambetov20a.html)
[58] Zharmagambetov A and Carreira-Perpiñán M A 2020 Ensembles of bagged tao trees consistently improve over random forests, adaboost and gradient boosting *Proc. 2020 ACM-IMS on Foundations of Data Science Conf., FODS'20* (Association for Computing Machinery) p 35
[59] CMS Collaboration 2024 The CMS statistical analysis and combination tool: COMBINE (arXiv:2404.06614)
[60] ROOT Collaboration 2012 HistFactory: a tool for creating statistical models for use with RooFit and RooStats *Technical Report* New York U
[61] Kassabov Z *et al* 2023 The top quark legacy of the LHC Run II for PDF and SMEFT analyses *J. High Energy Phys.* JHEP05(2023)205
[62] CMS Collaboration 2024 CMS open data guide (available at: https://cms-opendata-guide.web.cern.ch/)
[63] ATLAS Collaboration 2024 ATLAS open data portal (available at: https://atlas.cern/Resources/Opendata)
[64] ATLAS Collaboration 2023 Inclusive and differential cross-sections for dilepton $t\bar{t}$ production measured in $\sqrt{s} = 13$ TeV pp collisions with the ATLAS detector *J. High Energy Phys.* JHEP07(2023)141
[65] CMS Collaboration 2024 Differential cross section measurements for the production of top quark pairs and of additional jets using dilepton events from pp collisions at $\sqrt{s} = 13$ TeV (arXiv:2402.08486)
[66] NNPDF Collaboration 2017 Parton distributions from high-precision collider data *Eur. Phys. J.* C **77** 663
[67] Brivio I 2021 SMEFTsim 3.0 – a practical guide *J. High Energy Phys.* JHEP04(2021)073
[68] Skands P, Carrazza S and Rojo J 2014 Tuning PYTHIA 8.1: the Monash 2013 Tune *Eur. Phys. J.* C **74** 3024
[69] CMS Collaboration 2016 Event generator tunes obtained from underlying event and multiparton scattering measurements *Eur. Phys. J.* C **76** 155
[70] CMS Collaboration 2020 Extraction and validation of a new set of CMS PYTHIA8 tunes from underlying-event measurements *Eur. Phys. J.* C **80** 4
[71] Alwall J *et al* 2008 Comparative study of various algorithms for the merging of parton showers and matrix elements in hadronic collisions *Eur. Phys. J.* C **53** 473
[72] Cacciari M, Salam G P and Soyez G 2008 The anti-k_T jet clustering algorithm *J. High Energy Phys.* JHEP04(2008)063
[73] Cacciari M, Salam G P and Soyez G 2012 FastJet user manual *Eur. Phys. J.* C **72** 1896
[74] Elmer N, Madigan M, Plehn T and Schmal N 2023 Staying on Top of SMEFT-Likelihood Analyses (arXiv:2312.12502)
[75] CMS Collaboration 2023 Measurement of the tt^- charge asymmetry in events with highly Lorentz-boosted top quarks in pp collisions at s = 13 TeV *Phys. Lett.* B **846** 137703
[76] ATLAS Collaboration 2023 Evidence for the charge asymmetry in pp $\rightarrow t\bar{t}$ production at $\sqrt{s} = 13$ TeV with the ATLAS detector *J. High Energy Phys.* JHEP08(2023)077
[77] CMS Collaboration 2019 Measurement of the top quark polarization and $\boldsymbol{t\bar{t}}$ spin correlations using dilepton final states in proton-proton collisions at $\sqrt{s} = 13$ TeV *Phys. Rev.* D **100** 072002
[78] Bernreuther W, Heisler D and Si Z-G 2015 A set of top quark spin correlation and polarization observables for the LHC: Standard Model predictions and new physics contributions *J. High Energy Phys.* JHEP12(2015)026
[79] Butterworth J *et al* 2016 PDF4LHC recommendations for LHC Run II *J. Phys.* G **43** 023001
[80] CMS Collaboration 2021 CMS Open Data Workshop 2021 (available at: https://cms-opendata-workshop.github.io/2021-07-19-cms-open-data-workshop/)
[81] CMS Collaboration 2022 CMS Open Data Workshop 2022 (available at: https://cms-opendata-workshop.github.io/2022-08-01-cms-open-data-workshop/)
[82] CMS Collaboration 2023 CMS Open Data Workshop 2023 (available at: https://cms-opendata-workshop.github.io/2023-07-11-cms-open-data-workshop/)
[83] Friedman J H 2003 On multivariate goodness of fit and two sample testing *eConf* **C030908** THD002
[84] Lopez-Paz D and Oquab M 2018 Revisiting classifier two-sample tests (arXiv:1610.06545)
[85] Cowan G, Cranmer K, Gross E and Vitells O 2011 Asymptotic formulae for likelihood-based tests of new physics *Eur. Phys. J.* C **71** 1554 Cowan G, Cranmer K, Gross E and Vitells O 2013 *Eur. Phys. J.* C **73** 2501 (erratum)
[86] Wald A 1943 Tests of statistical hypotheses concerning several parameters when the number of observations is large *Trans. Am. Math. Soc.* **54** 426
[87] Dembinski H et al scikit-hep/iminuit (available at: http://dx.doi.org/10.5281/zenodo.3949207)
[88] Harris C R *et al* 2020 Array programming with NumPy *Nature* **585** 357
[89] Github repository 2024 Boosted parametric tree (available at: https://github.com/HephyAnalysisSW/BPT)