Machine learning for phase space integration with SHERPA

E Bothmann, T Janßen and S Schumann

Institut für Theoretische Physik, Georg-August-Universität Göttingen,
Friedrich-Hund-Platz 1, 37077 Göttingen, Germany

E-mail: `enrico.bothmann@uni-goettingen.de`

Abstract. Simulated event samples from Monte-Carlo event generators are a backbone of the LHC physics programme. However, for Run III, and in particular for the HL-LHC era, computing budgets are becoming increasingly constrained, while at the same time the push to higher accuracies is making event generation significantly more expensive. Modern ML techniques can help with the effort of creating such costly samples. Our focus here is to embed those techniques within the existing generator frameworks in an effort to increase the efficiency of certain performance bottlenecks. One of those bottlenecks is the sampling of the high-dimensional phase space of complex processes, for which a given distribution must be approximated as closely as possible. This is a generic problem, such that methods can be explored that have been developed in other fields of physics or even outside of it. In these proceedings we review studies to increase the phase space sampling efficiency using Normalising Flows and Nested Sampling in the Sherpa generator framework, with a particular focus on the unweighting efficiency. We also briefly comment on other approaches to address the given bottlenecks.

1. Introduction

Monte-Carlo event generators (MCEG) are one of the backbones of collider physics, be it for simulating background event samples, calibrating detector responses, or even for planning new colliders [1]. Given the basic building blocks of a Quantum Field Theory (QFT), such as its Lagrangian, they produce simulated scattering events. The main phases of the simulation are roughly ordered by the corresponding energy scales. The QFT matrix element for a few outgoing elementary particles is evaluated at the collision scale using perturbation theory. The parton shower dresses these particles with Bremsstrahlung emissions, ending with a large number of low-energy partons. These enter the hadronisation phase, in which these partons transition into hadrons, which in turn decay until only stable particles remain that are long-lived enough to enter the detector in the real world. In the simulated world, this is where the detector simulation would take over, which is however a component of the simulation pipeline which is separate from MCEG and will therefore not be discussed here.

General-purpose MCEG such as Pythia [2], Herwig [3, 4] and Sherpa [5], potentially combined with tree-level or loop matrix element generators, have become increasingly precise tools, incorporating higher orders of perturbation theory in the matrix element calculation and in the parton shower simulation and allowing to include higher multiplicity matrix elements, as required by experimental analyses and phenomenological studies. Especially since the discovery of the Higgs boson [6, 7], the focus of these studies is shifting to high-precision measurements and searches for increasingly subtle deviations potentially induced by New Physics.

This, alongside the very dynamic and general-purpose nature of the most-commonly used MCEG codes result in large computational costs of high-fidelity event samples, which are e.g. produced by ATLAS and CMS to serve many of their analyses. This is particularly true for standard-candle processes such as the production of a vector boson or a top-antitop quark pair in association with additional jet emissions. Their production makes up a significant part of the overall computing budget, and is projected to rise significantly in the upcoming HL-LHC era [8, 9].

This high cost naturally gives rise to the question if modern Machine Learning (ML) methods can be used to reduce it. Of course, *traditional* ML methods have been part of MCEG from the beginning. For example, adaptive algorithms are used to make sampling the phase space of the matrix element configurations more efficient, e.g. combining Vegas importance sampling [10] with an adaptive multi-channel Monte-Carlo [11]. However, the efficiency of this sampling, which even after optimisation can be very low for complex processes, combined with the high cost of the sampled integrand (i.e. the scattering matrix element) has been shown to be the bottleneck of the event generation with Sherpa for typical setups [12, 13]. A better efficiency would not only reduce the cost and ensure that the HL-LHC physics programme can succeed [8], but would also unlock new physics opportunities, by allowing for even better precision or higher multiplicities to be explored. Therefore, we will concentrate in these proceedings on the question whether modern ML techniques can be used to increase the efficiency.

Two clarifications are in order here. First note that the issue is indeed fairly generic and that we discuss methods that are relevant for any sampling/integration problem with similar characteristics. Secondly, we are here exclusively interested in applications, for which an insufficiently trained model only leads to a degradation of the achieved efficiency, but has no impact on the physics prediction of the simulation. The reason is that most applications of MCEG require faithful predictions, ones for which the physical input is precisely known and for which the uncertainty can be quantified. Making this a strict requirement sets our applications somewhat apart from attempts to replace (parts of) the MCEG toolchain with generative models [14, 15, 16, 17, 18], where great care must be taken to ensure that uncertainties are faithfully estimated and that the physics is correctly described not only in the bulk of the distributions of physical observables, but also in the tails, which are often important in the search for New Physics.

After discussing in more detail the computing challenges in question in Sec. 2 we describe the approaches of Normalising Flows in Sec. 3, and Nested Sampling in Sec. 4. After shortly visiting other approaches in Sec. 5 we conclude in Sec. 6.

2. Computing challenges in phase-space integration

The ATLAS Software and Computing HL-LHC Roadmap presents predictions showing a possible overshoot of the computing requirements with respect to the computing budget by up to a factor of four, depending on the assumptions made [9]. As reproduced in Fig. 1, MCEG (here labelled “EvGen”) can in one scenario become the largest consumer, using 20 % of the overall computing budget by ATLAS. While this finding is partially attributable to the assumed improvements in detector simulation efficiency in the given scenario, the actual numbers are not far from this projection, and MCEG computing usually accounts for 5 %–20 % of the overall ATLAS and CMS computing budgets. It is therefore vital to make the toolchain ready for the increased requirements of the HL-LHC era. When we look into where computing time is spent for generating a typical high-statistics unweighted event sample using Sherpa (e.g. Z production in association with up to two additional jets at NLO in QCD and up to 3 more jets at LO), as given in Fig. 2, we find that more than two thirds of the overall computing time is spent

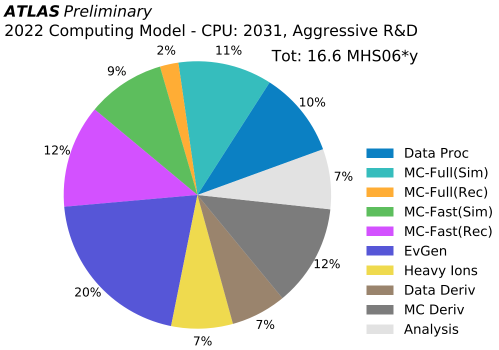

Figure 1. ATLAS computing budget as projected for 2031, taken from [9]. MCEG (here: "EvGen") amounts to 20 % of the overall computing in the given scenario.

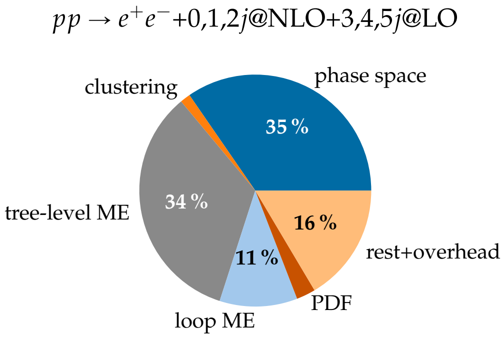

Figure 2. Computing time for the different parts of a MCEG simulation of a typical high-statistics sample used by experiments. Figure taken from [13].

evaluating tree-level matrix elements (34 %) and generating phase-space points (35 %).

While improving the performance of these two bottlenecks can be done independently and will shrink the respective pieces of the pie chart, we can actually shrink both pieces at the same time if we find a way to increase the unweighting efficiency $\epsilon = \langle N_{\text{trials}}\rangle^{-1}$, where $\langle N_{\text{trials}}\rangle$ is the average number of trial Monte-Carlo points per accepted point/event in the unweighting procedure.[1] For $Z+$jets production at LO, the phase-space efficiency of Sherpa drops from $\mathcal{O}(10^{-1})$ for one additional jet to $\mathcal{O}(10^{-4})$ for five additional jets [19]. The reason for these rather low efficiencies is threefold. The integral to be evaluated,

$$\sigma_{pp\to X_n} = \sum_{ab} \int \mathrm{d}x_a \mathrm{d}x_b\, \mathrm{d}\Phi_n\; f_a(x_a,\mu_F^2) f_b(x_b,\mu_F^2)\; |\mathcal{M}_{ab\to X_n}|^2\; \Theta_n(p_1,\ldots,p_n), \tag{1}$$

contains with $f_a f_b |\mathcal{M}|^2$ (the parton density functions of the incoming protons times the matrix element squared) a multi-modal, wildly fluctuating function with a number of (integrable) singularities. Moreover, it is subject to non-trivial acceptance cuts $\Theta_n(p_1,\ldots,p_n)$ throughout phase space, introducing discontinuities. And finally, the number of dimensions $\dim[\Phi_n] = 3n-4$ is quite high, e.g. $\dim[\Phi_7] = 17$ for the aforementioned Z plus five jet production.

It is important to note at this point that MCEG typically use knowledge about the physical structure of $f_a f_b |\mathcal{M}|^2$ to construct channels in a multi-channel Monte-Carlo, mapping out prominent features/singularities. While not all features are known, or can be easily determined in a generic program, this procedure has been invaluable to render the integration problem computationally tractable. Each channel is then further optimised using an adaptive remapping of the sampling space into the latent space with the Vegas algorithm [10]. This works well as long as the integrand can be factorised along the dimensions of the integration volume, which is a limitation of the Vegas algorithm. But even after the separation of the channels, this is not the case here. A modern ML method might be able to provide a more flexible mapping. Based on this one could decide to use a single channel and sample points globally. Instead

[1] Note that the unweighting efficiency is only one of the figures of merit, the other one would be the estimator of the variance, $\mathrm{Var} = (\langle w^2\rangle - \langle w\rangle^2)/N$, where w denotes the weight of the individual events. A small variance is better, and usually—but not necessarily—comes with a better unweighting efficiency. Since large sample production makes use of unweighting which leads to smaller data sets, we concentrate on the unweighting efficiency here. For a more detailed discussion, including the effect of the optimisations on the variance, we refer to the original publications.

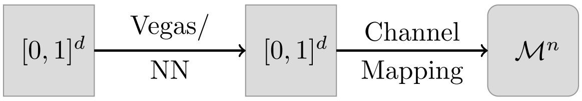

Figure 3. For each channel of the multi-channel Monte Carlo, (uniform) random numbers from a latent space are first mapped to the intermediary latent space of a phase-space mapping, which then maps into the target space of the physical momenta entering the n-particle matrix element $\mathcal{M}^n$.

one can also leave the multi-channel structure in place to simplify the problem and just replace or complement Vegas. This allows one to profit from the physics knowledge built into the multi-channel construction. We explore this possibility in the next section.

3. Normalising Flows

Figure 3 shows how (uniform) random numbers of a d-dimensional unit cube latent space are first mapped using an optimiser such as Vegas to an intermediary unit cube latent space, which in turn is mapped by a channel mapping to the physical momenta. There is one such chain of mappings per phase-space channel of the multi-channel Monte Carlo.

If we want to use the strategy laid out in the previous section to use modern ML methods as a replacement for the first (Vegas) mapping, the provided mapping must be bijective: Surjectivity is needed to guarantee full phase-space coverage, and injectivity is required to be able to calculate the inverse Jacobian for each phase-space point, which is an input for the phase-space weight of the generated event. Moreover, the unweighting efficiency must be at least as good as what we can achieve with Vegas, and the generation of points and their phase-space weights should not be too expensive computationally. In particular the scaling with the number of dimensions should be under control.

A Normalising flow (NF) is a possible candidate. NFs have been applied exploratively to several problems in high-energy physics and adjacent fields [20, 21, 22, 23, 24, 25, 26, 27, 19]. Bijectivity is guaranteed by construction, and the evaluation of the inverse Jacobian is typically cheap with a computational complexity of $\mathcal{O}(d)$, where d is the number of dimensions. At the same time, NFs allow for expressive non-linear variable transformations.

In Fig. 4 (left), we show the collected results of [20] and [22] for using NFs as a Vegas replacement to sample the phase space of various processes. Details on the respective methodologies and parameter choices are beyond the scope of these proceedings, and we refer the reader to the two original publications. The y axis shows the gain in efficiency, i.e. $\epsilon_{\mathrm{NF}}/\epsilon_{\mathrm{VEGAS}}$, while the x axis gives the number of kinematical dimensions d (ignoring hyperparameters, e.g. to choose a particular channel). The circles show results from [22] for vector-boson plus jet production processes at the LHC, while the triangles show results from [20] for the decay of a top quark t, the production of a top-antitop quark pair at a lepton-lepton collider, and gluonic scattering processes at a fixed centre-of-mass energy with $n = 3, 4$ gluons in the final state. One finds promising gains of more than 3, but these gains quickly vanish for higher numbers of dimensions, until eventually the gain drops below 1, i.e. using Vegas becomes more efficient than using NFs.

In Fig. 4 (right), we show more gain factors, which have been reported in [19]. The new simple phase-space generator Chili is used, and it is configured to only use the single phase-space mapping

$$\mathrm{d}x_a\,\mathrm{d}x_b\,\mathrm{d}\Phi_n(a,b;1,\ldots,n) = \frac{2\pi}{s}\left[\prod_{i=1}^{n-1}\frac{1}{16\pi^2}\,\mathrm{d}p_{i,\perp}^2\,\mathrm{d}y_i\frac{\mathrm{d}\phi_{\mathrm{i}}}{2\pi}\right]\mathrm{d}y_n, \tag{2}$$

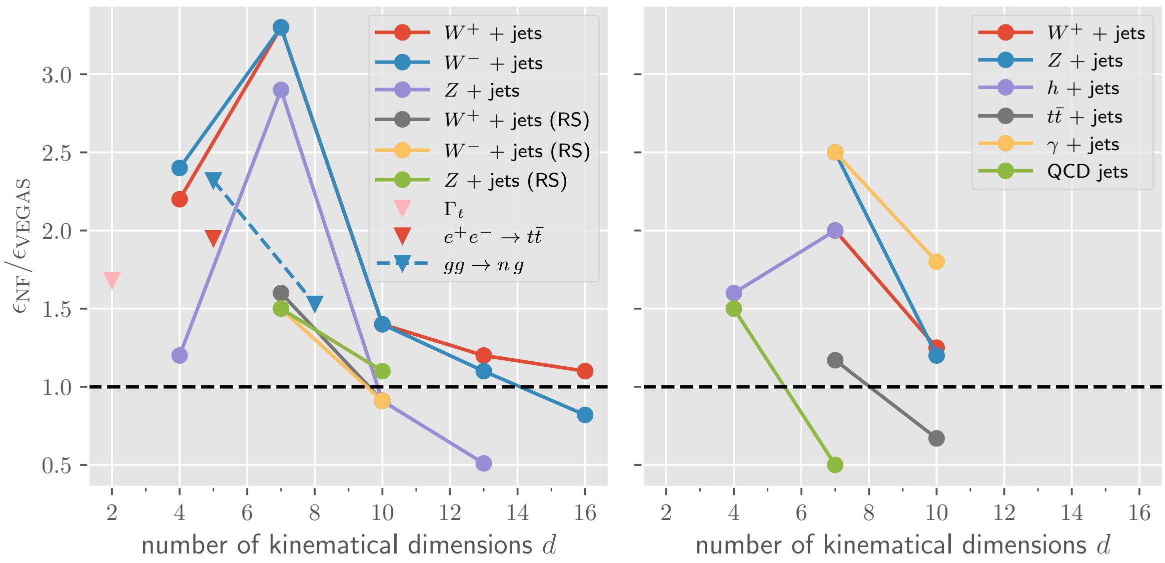

Figure 4. The ratio of unweighting efficiencies ϵ using Normalising Flows and VEGAS versus the number of kinematical dimensions for various physics processes. The data is taken from [22] (left, circles), [20, 28] (left, triangles) and [19] (right).

such that no hyperparameters are required to choose between different channels. Despite this simplification, there is no improvement to be found with respect to the older results discussed above. However, it should be stressed that this study is only a proof-of-principle for the given interfaces to the NF training frameworks i-FLOW and MADNIS [21, 24], and no attempt has been made to optimise the training.

In all cases in Fig. 4, we eventually find decreasing gains with increasing dimensionality. This is discouraging, since speed improvements are most relevant for higher particle multiplicities. However, it is likely that this is at least partially attributable to a lack of training. The bottleneck of the training for high dimensionalities has in all cases been the serial evaluation of the matrix elements on the CPU. The recent advances of GPU-accelerated matrix-element generators [29, 30, 31, 32] are therefore a promising prospect, and it is worth revisiting the Normalising Flows when these generators become available.

4. Nested Sampling

We now explore the application of Nested Sampling (NS) to our problem of phase-space integration, as reported in [33]. NS is an iterative Bayesian inference algorithm [34]. In the first iteration step, a number of points is uniformly sampled. Then, the point corresponding to the lowest value of the target function is identified and replaced by another point under the constraint that it gives a higher value. After repeating this procedure until an appropriate termination criterion has been reached, the union of the discarded points and the remaining live points forms a representative sample of the target distribution. Existing applications range from cosmology through statistical thermodynamics to material science, and there is a range of tools that implement the central sampling algorithm and/or perform auxiliary tasks. One of these tools is POLYCHORD [35], and it is used in [33], along with the `anesthetic` tool to join NS samples [36]. We refer to [33] for more details, in particular for the hyperparameters used to steer POLYCHORD.

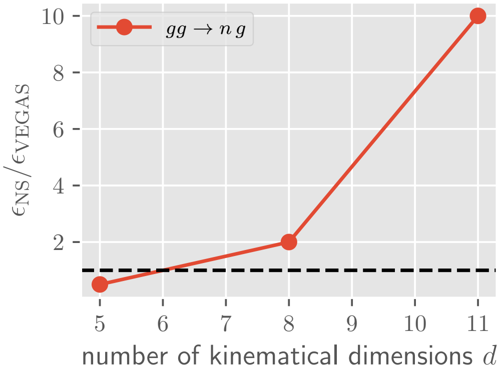

Figure 5. The ratio of unweighting efficiencies ϵ using Nested Sampling and Vegas versus the number of kinematical dimensions for gluon scattering with n gluons in the final state. The data is taken from [33]. Note that in contrast to Fig. 4 here Vegas refers to a stand-alone mapping as a single channel without prior knowledge.

Note that the method as used in [33] uses no prior knowledge about the target, i.e. the starting point for the iteration is a flat prior. In contrast to the NF applications in Sec. 3, this means that no channel mappings and in particular no multi-channel Monte Carlo is used. Hence, the NS algorithm must perform the full adaptation to the multi-modal and wildly fluctuating target function. The posterior of the iterative inference should match the target distribution.

Figure 5 shows the gains with respect to Vegas optimisation that were achieved, in terms of the unweighting efficiency ϵ. The gains are rather high, reaching about 10 for the $gg \to 5g$ process, with the number of dimensions being 11. As can be seen in [33], the results from the Vegas efficiency decline rapidly, while the NS efficiency remains almost constant. This is a very promising result, in particular with regard to the favourable scaling.

5. Other approaches

There are three other approaches to address the event generation bottleneck described in Sec. 2 that we would like to highlight here. None of these in itself leads to an improvement of the unweighting efficiency, but instead they focus on the reduction of the computational effort to calculate (unweighted) events.

The first one has already been mentioned towards the end of Sec. 3, namely the use of GPU acceleration for matrix-element evaluation and possibly also phase-space generation. This has the potential to drastically decrease the computation time required per phase-space point, such that a low efficiency becomes less of an issue. Porting both the matrix element and the phase space is highly desirable to avoid the need to copy phase-space points to the GPU. As long as the memory requirements are under control, the independence of the Monte-Carlo points means that the use of an accelerator such as a GPU is a perfect fit, allowing to parallelise the event generation on the device, instead of generating events serially on the CPU. GPU-accelerated matrix-element generators are currently an object of active research and development [29, 30, 31, 32].

The second approach is the use of pilot runs in existing Monte-Carlo frameworks. These make sure that as much work as possible is postponed until after the unweighting, i.e. is only done for accepted events. In Sherpa, a very significant speed-up of about a factor of 40 has recently been achieved for a typical ATLAS V+jets setup, mostly but not exclusively by postponing the calculation of variation weights needed for the evaluation of theory uncertainties in that way [13].

The third optimisation strategy that is actively explored is the use of approximate and computationally cheap matrix-element models for trial events, so-called surrogates. When surrogates are used, the unweighting becomes a two-step procedure. First the trial events are unweighted against the nominal weight maximum. If this is accepted, a second accept/reject step is done against the ratio to the nominal matrix element. This procedure ensures that the distribution of the final sample of unweighted events is correct. Because the computationally expensive matrix element is only evaluated if the first unweighting step is accepted, the time improvement can be significant. Since a low efficiency of the second unweighting step would decrease the overall unweighting efficiency, it must be ensured that the approximate model is as faithful as possible to the exact matrix element. Recently, this technique of "surrogate unweighting" has been explored using deep neural networks [37]. Speed gain factors between 2 and 10 have been reported. In a follow-up study, a more sophisticated neural network model constructed on the basis of the factorisation properties of QCD matrix elements, as proposed in Ref. [38], has been applied to multi-jet production [39]. While this is currently limited to colour-summed matrix elements, the speed gain factors have been found to be very significant, between 16 and 350. These results are very promising, and the technique provides an excellent use case for sophisticated modern ML inspired ME emulators [40, 41].

6. Conclusions

We have presented several approaches to address the computational bottleneck of Monte Carlo event generation in the context of high-energy physics, namely the expensive evaluation of matrix elements and the low efficiency of the phase-space sampling. The underlying problem to efficiently integrate and sample from a complex multi-modal distribution with non-trivial discontinuities is very generic and of interdisciplinary relevance, and there is possible crosstalk to other fields such as Machine Learning, lattice field theory, cosmology and industry. The two approaches mainly discussed in these proceedings, Normalising Flows and Nested Sampling, have indeed been developed outside of High-Energy Physics and are now being explored in this context.

Normalising Flows promise an improvement in the efficiency, since they can be used as a more flexible replacement for the conventional Vegas optimiser. While the proof-of-principle studies so far indeed showed improvements for lower dimensionalities, the scaling behaviour seemed less favourable. Eventually, the conventional approach proves to be hard to improve upon. This however could be an effect of insufficient training, which is very expensive when the target function is evaluated serially on a CPU. GPU-accelerated matrix elements are highly desirable for further research into the use of Normalising Flows.

Nested Sampling has been shown to give good improvements and a favourable scaling behaviour with the number of dimensions for the test case of gluon scattering. It will be interesting to see if follow-up studies can successfully generalise these findings to other and/or more complex processes.

We also briefly discussed other approaches which alleviate the issue of the low phase-space efficiency by decreasing the time needed to evaluate a given phase-space point, namely the already mentioned use of GPU-accelerated matrix elements, and the use of pilot runs and surrogate unweighting. All of these show great promise, and pilot runs have already been integrated into the existing event generation pipeline.

Making at least a subset of the above methods production-ready and integrating them into the existing pipeline is indeed required to fully utilise the upcoming HL-LHC era.

Acknowledgments

We acknowledge support from BMBF (contract 05H21MGCAB) and funding by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 456104544; 510810461.

References

[1] Buckley A *et al.* 2011 *Phys. Rept.* **504** 145–233 (*Preprint* 1101.2599)

[2] Bierlich C *et al.* 2022 (*Preprint* 2203.11601)

[3] Bahr M *et al.* 2008 *Eur. Phys. J. C* **58** 639–707 (*Preprint* 0803.0883)

[4] Bellm J *et al.* 2016 *Eur. Phys. J. C* **76** 196 (*Preprint* 1512.01178)

[5] Bothmann E *et al.* (Sherpa) 2019 *SciPost Phys.* **7** 034 (*Preprint* 1905.09127)

[6] Aad G *et al.* (ATLAS) 2012 *Phys. Lett. B* **716** 1–29 (*Preprint* 1207.7214)

[7] Chatrchyan S *et al.* (CMS) 2012 *Phys. Lett. B* **716** 30–61 (*Preprint* 1207.7235)

[8] Yazgan E *et al.* (HSF Physics Event Generator WG) 2021 (*Preprint* 2109.14938)

[9] ATLAS Collaboration 2022 ATLAS Software and Computing HL-LHC Roadmap Tech. rep. CERN Geneva

[10] Lepage G P 1978 *J. Comput. Phys.* **27** 192

[11] Kleiss R and Pittau R 1994 *Comput. Phys. Commun.* **83** 141–146

[12] Höche S, Prestel S and Schulz H 2019 *Phys. Rev. D* **100** 014024 (*Preprint* 1905.05120)

[13] Bothmann E, Buckley A, Christidi I A, Gütschow C, Höche S, Knobbe M, Martin T and Schönherr M 2022 *Eur. Phys. J. C* **82** 1128 (*Preprint* 2209.00843)

[14] Stienen B and Verheyen R 2021 *SciPost Phys.* **10** 038 (*Preprint* 2011.13445)

[15] Butter A, Plehn T and Winterhalder R 2019 *SciPost Phys.* **7** 075 (*Preprint* 1907.03764)

[16] Di Sipio R, Giannelli M F, Haghighat S K and Palazzo S 2019 *Journal of High Energy Physics* **2019** 110 (*Preprint* 1903.02433)

[17] Otten S, Caron S, de Swart W, van Beekveld M, Hendriks L, van Leeuwen C, Podareanu D, Ruiz de Austri R and Verheyen R 2021 *Nature Commun.* **12** 2985 (*Preprint* 1901.00875)

[18] Choi S and Lim J H 2021 *J. Korean Phys. Soc.* **78** 482–489 (*Preprint* 2102.11524)

[19] Bothmann E, Childers T, Giele W, Herren F, Hoeche S, Isaacsson J, Knobbe M and Wang R 2023 (*Preprint* 2302.10449)

[20] Bothmann E, Janßen T, Knobbe M, Schmale T and Schumann S 2020 *SciPost Phys.* **8** 069 (*Preprint* 2001.05478)

[21] Gao C, Isaacson J and Krause C 2020 *Mach. Learn. Sci. Tech.* **1** 045023 (*Preprint* 2001.05486)

[22] Gao C, Höche S, Isaacson J, Krause C and Schulz H 2020 *Phys. Rev. D* **101** 076002 (*Preprint* 2001.10028)

[23] Winterhalder R, Magerya V, Villa E, Jones S P, Kerner M, Butter A, Heinrich G and Plehn T 2022 *SciPost Phys.* **12** 129 (*Preprint* 2112.09145)

[24] Heimel T, Winterhalder R, Butter A, Isaacson J, Krause C, Maltoni F, Mattelaer O and Plehn T 2022 (*Preprint* 2212.06172)

[25] Jinno R, Kälin G, Liu Z and Rubira H 2022 (*Preprint* 2209.01091)

[26] Badger S *et al.* 2022 (*Preprint* 2203.07460)

[27] Singha A, Chakrabarti D and Arora V 2023 *Phys. Rev. D* **107** 014512 (*Preprint* 2207.00980)

[28] Janßen T and Schumann S 2023 *Journal of Physics: Conference Series* **2438** 012144

[29] Bothmann E, Giele W, Hoeche S, Isaacson J and Knobbe M 2021 (*Preprint* 2106.06507)

[30] Valassi A, Roiser S, Mattelaer O and Hageboeck S 2021 *EPJ Web Conf.* **251** 03045 (*Preprint* 2106.12631)

[31] Valassi A, Childers T, Field L, Hageboeck S, Hopkins W, Mattelaer O, Nichols N, Roiser S and Smith D 2022 *PoS* **ICHEP2022** 212 (*Preprint* 2210.11122)

[32] Bothmann E, Isaacson J, Knobbe M, Höche S and Giele W 2022 *PoS* **ICHEP2022** 222

[33] Yallup D, Janßen T, Schumann S and Handley W 2022 *Eur. Phys. J. C* **82** 8 (*Preprint* 2205.02030)

[34] Skilling J 2006 *Bayesian Analysis* **1** 833–859

[35] Handley W J, Hobson M P and Lasenby A N 2015 *Monthly Notices of the Royal Astronomical Society* **453** 4385–4399

[36] Handley W 2019 *J. Open Source Softw.* **4** 1414 (*Preprint* 1905.04768)

[37] Danziger K, Janßen T, Schumann S and Siegert F 2022 *SciPost Phys.* **12** 164 (*Preprint* 2109.11964)

[38] Maître D and Truong H 2021 *JHEP* **11** 066 (*Preprint* 2107.06625)

[39] Janßen T, Maître D, Schumann S, Siegert F and Truong H 2023 (*Preprint* 2301.13562)

[40] Aylett-Bullock J, Badger S and Moodie R 2021 *JHEP* **08** 066 (*Preprint* 2106.09474)

[41] Badger S, Butter A, Luchmann M, Pitz S and Plehn T 2022 (*Preprint* 2206.14831)