Unweighted event generation for multi-jet production processes based on matrix element emulation

T Janßen[1*], D Maître[2], S Schumann[1], F Siegert[3] and H Truong[2]

[1]Institut für Theoretische Physik, Georg-August-Universität Göttingen, DE
[2]Institute for Particle Physics Phenomenology, Department of Physics, Durham University, UK
[3]Institut für Kern- und Teilchenphysik, TU Dresden, DE
[*]Speaker

E-mail: timo.janssen@theorie.physik.uni-goettingen.de

Abstract. The generation of unit-weight events for complex scattering processes presents a severe challenge to modern Monte Carlo event generators. Even when using sophisticated phase-space sampling techniques adapted to the underlying transition matrix elements, the efficiency for generating unit-weight events from weighted samples can become a limiting factor in practical applications. Here we present the combination of a two-staged unweighting procedure with a factorisation-aware matrix element emulator using neural networks which we make accessible in the Sherpa event generation framework. The algorithm can significantly accelerate the unweighting process, while it still guarantees unbiased sampling from the correct target distribution. We apply, validate and benchmark the approach for partonic channels contributing to the high-multiplicity LHC production processes $Z + 4, 5$ jets and $t\bar{t} + 3, 4$ jets, where we find speed-up factors between 16 and 350.

1. Introduction

The high luminosities achieved by the LHC lead to a need for large numbers of simulated events for physics analyses. These often address rather rare and complex scattering processes that need to be described with sufficient theoretical accuracy. Modern Monte Carlo event generators like Herwig [1, 2], Pythia [3, 4] and Sherpa [5, 6] are up to the task but the resource requirements can become substantial. The upcoming HL-LHC will aggravate this situation, especially since current projections indicate that the growth in computational budget will not match the growth in computational demand [7]. Accordingly, the efficiency of event generation plays an important role. Machine learning has become a promising tool to address this challenge.

Here, we focus on the hard interaction. The time-consuming detector simulation that post processes the output of the event generator makes it desirable to produce unit weight events using rejection sampling. However, for large parton multiplicities the efficiency can be very low due to the high phase-space dimensionality. While other methods aim at improving the unweighting efficiency [8–15], we here consider a complementary strategy. The underlying idea is that the overall event generation time can be reduced by reducing the number of calls to the matrix element (ME), which is typically the most expensive part of the calculation, especially at high multiplicity.

We report on the results of combining the findings of two previous studies. In Ref. [16] the authors presented a factorisation-aware surrogate model for QCD MEs. We briefly describe the

model in Sec. 2 and explain how we extended it to hadronic initial states and massive partons. Subsequently, in Sec. 3, we describe the two-stage surrogate unweighting algorithm of Ref. [17]. It makes use of a fast and accurate ME surrogate to speed up unweighted event generation while maintaining unbiasedness by introducing a second unweighting step. Our results of applying the combination of the two ideas to LHC production processes are shown in Sec. 4. We end with our conclusions in Sec. 5.

2. Matrix element emulation with factorisation-aware neural networks

The factorisation properties of QCD matrix elements in their soft and collinear limits can be written as

$$|\mathcal{M}_{n+1}|^2 \to |\mathcal{M}_n|^2 \otimes \mathbf{V}_{ijk}\,, \tag{1}$$

where the $(n+1)$-body ME $|\mathcal{M}_{n+1}|^2$ factorises into a reduced ME in n-body phase space and a singular factor $\mathbf{V}_{ijk}$. In the dipole formalism introduced by Catani and Seymour in Ref. [18] all divergences are captured by dipole functions D_{ijk} which can be used to build an ansatz for the colour and helicity summed ME:

$$|\mathcal{M}_{n+1}|^2 \approx \sum_{\{ijk\}} C_{ijk} D_{ijk}\,. \tag{2}$$

The indices i, j and k label the three partons involved in the splitting process. Since the coefficients C_{ijk} are more well behaved than the ME itself, they are better suited as a learning target for a NN. The NN does not need to learn the singular behaviour in infrared regions of phase space as these are described by the dipole functions. This makes it possible to accurately fit a ME over the whole sampled phase space with a single, comparably simple NN.

When the model was originally introduced in Ref. [16], the authors considered jet production processes in e^+e^- collisions where only one type of dipole is needed. In Sec. 4 we consider processes with QCD initial states and massive partons in the final state. Therefore two extensions of the model are necessary. First, we add the three types of dipoles where the emitter or spectator is in the initial state. Secondly, we include the massive forms of the dipole functions. Since they are more complex than the massless versions, we use only the minimal number of massive dipoles for each partonic channel.

The inputs to the NN are the 4-momenta p of the external particles, the dipole variables y_{ijk} and the kinematic invariants s_{ij} of all pairs of external particles. The latter two are not independent since they are derived from the momenta. After preprocessing of the inputs using normalisation and logarithmic scaling, they are processed by four hidden layers with 128 nodes each. The architecture of the NN is illustrated in Fig. 1. It uses the same number of hidden layers and nodes as the less accurate model of Ref. [17] and thus the evaluation time is similar. It is only slightly increased due to the more involved pre- and post-processing of the inputs and outputs.

To demonstrate the quality of the model and the extension to QCD initial states we show in Fig. 2 the truth-to-prediction ratio $|\mathcal{M}|^2_{\text{true}}/|\mathcal{M}|^2_{\text{pred}}$ between the true and the emulated ME against the true ME for the process $gg \to e^-e^+gg\bar{d}d$ contributing to $Z+4$ jets production at the LHC. It is more challenging than the examples considered in Ref. [16], due to the higher dimensionality and the more complicated singularity structure. This manifests itself in the fact that the values of $|\mathcal{M}|^2$ extend over 40 orders of magnitude and leads to a less accurate approximation. However, for the vast majority of points the ratio is close to one. At large values of $|\mathcal{M}|^2_{\text{true}}$ the deviations are mostly small. We find somewhat larger deviations at small values of $|\mathcal{M}|^2_{\text{true}}$. However, these contribute only little to the total cross section.

As shown in Ref. [20], the method can be extended to loop MEs using antenna functions. Both tree-level and one-loop antennae are needed but since the same singularity structure is described by fewer antennae than dipoles, the number of contributing terms is comparable.

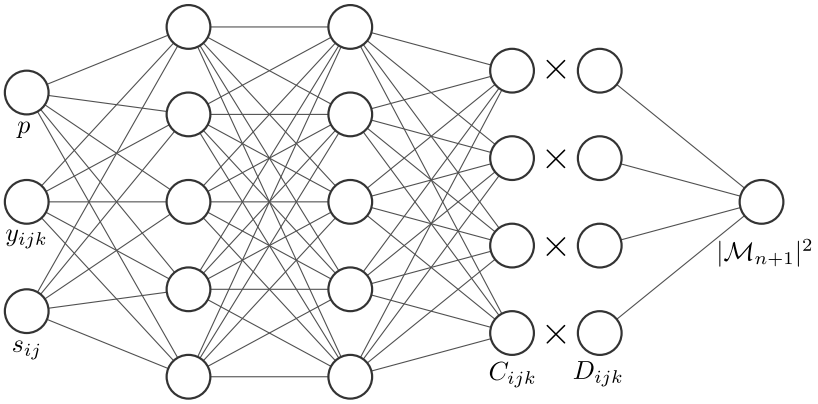

Figure 1: A simplified sketch of our neural network emulator showing input variables, hidden layers, and outputs C_{ijk}. Figure taken from Ref. [19].

3. Event unweighting with matrix element surrogates

Let us assume that we have a Monte Carlo sampling algorithm at hand that produces events with weights w such that the cross section of the target process can be estimated as the mean of the weights. The usual procedure for generating unweighted events would be to use rejection sampling. The approach of our surrogate unweighting algorithm is similar but employs two rejection sampling steps. In the first step, we draw a trial event from the proposal distribution and accept or reject the event based on the surrogate weight s. For this, we uniformly draw a random number R_1 from $[0, 1)$ and accept the event if $s > R_1 \cdot w_{\max}$, where $w_{\max}$ is the predetermined weight maximum. Only if an event has been accepted we calculate the true event weight w and determine the ratio $x = w/s$. For the second rejection sampling step we generate a second random number R_2 and accept the event if $x > R_2 \cdot x_{\max}$. After repeating the procedure several times, the set of events that have been accepted in both steps form an unweighted event sample. Since the computationally expensive ME only has to be evaluated for the events that have been accepted in the first step, resources can be saved provided that the surrogate is fast and that the acceptance rate is much higher in the second step than in the first one. The requirements for the latter are that the unweighting efficiency of the standard method is fairly low and the surrogate is highly accurate.

There is a subtlety concerning the weight maxima $w_{\max}$ and $s_{\max}$. They have to be determined from an event sample of finite size. As a consequence, it is not guaranteed that they are the actual maximum values. If, during unweighted event generation, an event with a weight exceeding the respective predetermined maximum is encountered, an overweight has be assigned to it. The final weight, after both accept/reject steps, is then given by $\tilde{w} = \max(1, s/w_{\max}) \cdot \max(1, x/x_{\max})$ and the event sample is *partially unweighted*. As long as the overweights are taken account of, the distribution of events is statistically correct. To keep the effects of overweights under control, we use the following method. Based on an initial event sample, we set $w_{\max}$ and $x_{\max}$ such that the overweights do not contribute more than 0.1 % to the total cross section. Note that the same method is generically used by Sherpa.

4. Results

We train the model on 1M events generated by Sherpa using TensorFlow [21] and Keras [22]. As a loss function we use the mean squared error. The trained model gets exported in the Onnx format [23]. For model evaluation during unweighted event generation we use the Onnx Runtime [24] in a customised version of Sherpa-2.2. The true colour summed MEs get evaluated

Table 1: Effective gain factors for different processes. For comparison the results obtained using the *naive* neural network surrogate model from Ref. [17] are given. Note that the naive model includes the phase space weight while the dipole model learns the matrix element weight only.

Process		$gg \to e^+e^-ggd\bar{d}$	$gg \to e^+e^-gggd\bar{d}$	$u\bar{u} \to t\bar{t}d\bar{d}g$	$gg \to t\bar{t}ggg$	$ug \to t\bar{t}gggu$
f_{eff}	naive	2	26	1	3	11
	dipole	16	269	20	61	354

by Sherpa's built-in ME generator Amegic [25]. Throughout the event generation loop all calculations are done sequentially on a single CPU core.

As a first assessment of the quality of the factorisation-aware surrogate, we show in Fig. 3 the distribution of the ratio $x = w/s$ between the true event weights and the surrogate weights for the partonic process $gg \to e^-e^+ggd\bar{d}$. For comparison, we also present the results obtained using the simpler non-factorisation-aware (naive) surrogate model from Ref. [17]. We note that the naive model approximates the full event weight while the factorisation-aware model solely learns the ME weight and has to be augmented by the true phase-space weight, which is cheap to evaluate though. In Fig. 3 it can be seen that the dipole model achieves a narrower distribution of weights and a smaller $x_{\max}$, 2.6 compared to 41.5. This means that for the bulk of the events the dipole model gives a better approximation and that more events will be accepted in the second unweighting step.

As the main figure of merit we use the effective gain factor defined as the ratio between the average time it takes to generate an unweighted event using the standard method and the average time it takes using the two-stage surrogate unweighting algorithm:

$$f_{\text{eff}} \coloneqq \frac{T_{\text{standard}}}{T_{\text{surrogate}}} \,. \tag{3}$$

It includes the evaluation times of the matrix element and phase space weights as well as the time that is spent on rejected events.

We apply our method to partonic multijet processes in order to measure its performance. In particular, we consider various partonic channels contributing to $Z + \{4, 5\}$ jets and $t\bar{t} + \{3, 4\}$ jets production in proton-proton collisions at $\sqrt{s} = 13\,\text{TeV}$. For these, the number of relevant dipole terms ranges from 40 to 138. The results for the effective gain factors are shown in Tab. 1. For comparison, we also present the results obtained using the simpler non-factorisation-aware (naive) surrogate model from Ref. [17]. For the dipole model we achieve gain factors between 16 and 354. In all cases, the gains are significantly larger than with the naive model, ranging between 1 and 26. The largest gains are obtained for the processes with the highest multiplicity where the computationally most demanding MEs are encountered.

At large parton multiplicity it becomes attractive to sample the colours of the external partons instead of summing over them, see for example Ref. [26]. It is thus interesting to consider an extension of our method to colour-sampled amplitudes. We note that our model ansatz Eq. (2) is tailored to colour-summed MEs. However, as a naive approach it is possible to feed the colour assignments as additional inputs to the model and let the NN learn the underlying structure. We tested this for the processes $gg \to e^-e^+ggd\bar{d}$ and $gg \to t\bar{t}ggg$ using Sherpa's built-in ME generator Comix [27] but we were not able to achieve net gains. Several reasons for this can be identified. One being that the addition of the colour assignments increases the dimensionality of the problem and thus the approximation of the model worsens. Furthermore, the evaluation of a single colour amplitude is much faster than the evaluation of a whole colour summed ME. As a consequence, the emulated amplitude is not that much faster than the true one. In addition,

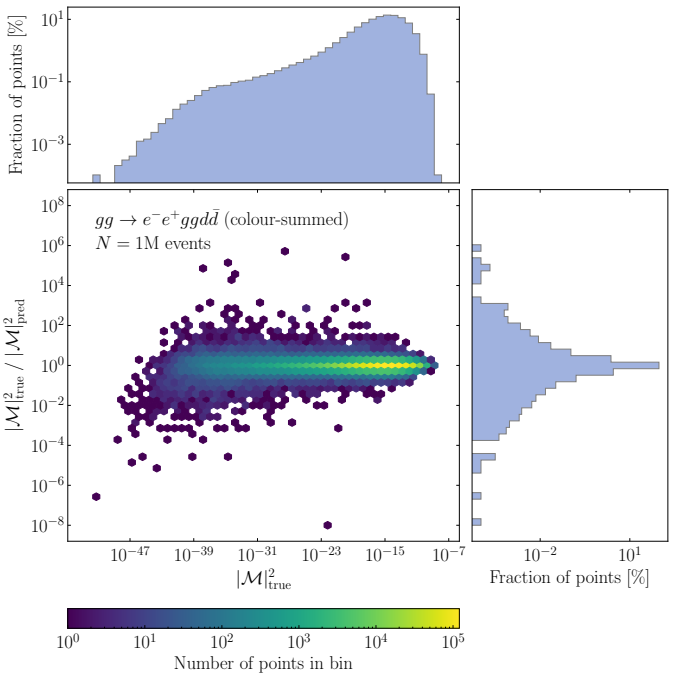

Figure 2: 2d histogram of the matrix element truth-to-prediction ratio for the $Z + 4j$ process $gg \to e^- e^+ ggd\bar{d}$. Along the axes, we plot the marginal distributions of the matrix element (top), and the truth-to-prediction ratio (right). High population bins are illustrated as yellow, while low population bins, down to single points, are depicted in purple. Figure taken from Ref. [19].

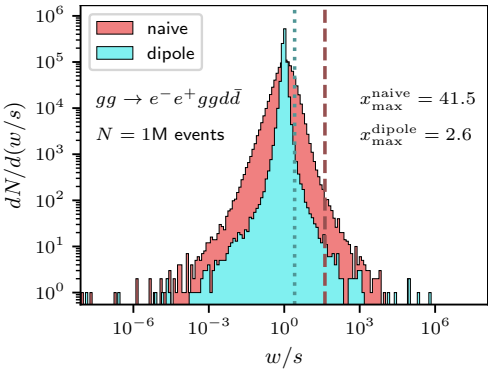

Figure 3: Ratio distributions of exact weights and their surrogate for the channel $gg \to e^+ e^- ggd\bar{d}$ ($Z + 4$ jets) using the factorisation-aware emulation of the matrix-element weight (dipole) and the combined matrix-element and phase-space weight from Ref. [17] (naive). Figure taken from Ref. [19].

this means that the evaluation of the phase-space weight contributes more significantly to the event generation time.

5. Conclusions

In these proceedings we reported on applying the combination of the factorisation-aware ME emulator of Ref. [16] and the unbiased two-stage unweighting algorithm of Ref. [17] to LHC multijet production processes. We found that the surrogate model is highly accurate for the colour-summed MEs of the considered processes. It was necessary to extend the model to hadronic initial states and massive external partons. Furthermore, we achieved large gain factors when using the model for unweighted event generation. We have taken the first steps towards transferring the accomplishments to colour-sampled amplitudes. Future work is necessary to complete this, especially by developing optimised surrogate models.

Acknowledgments

We thank the organisers for the exciting and pleasant conference. The work of SS and TJ was supported by BMBF (contract 05H21MGCAB) and Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - project number 456104544. FS's research was supported by the German Research Foundation (DFG) under grant No. SI 2009/1-1. DM's research was supported by STFC under grant ST/X003167/1.

References

[1] Corcella G, Knowles I G, Marchesini G, Moretti S, Odagiri K, Richardson P, Seymour M H and Webber B R 2001 *JHEP* **01** 010

[2] Bellm J *et al.* 2016 *Eur. Phys. J. C* **76** 196

[3] Sjöstrand T, Mrenna S and Skands P Z 2006 *JHEP* **05** 026

[4] Sjöstrand T *et al.* 2015 *Comput. Phys. Commun.* **191** 159–177

[5] Gleisberg T, Höche S, Krauss F, Schönherr M, Schumann S, Siegert F and Winter J 2009 *JHEP* **02** 007

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

[7] Collaboration A 2022 ATLAS Software and Computing HL-LHC Roadmap Tech. Rep. CERN-LHCC-2022-005, LHCC-G-182 CERN Geneva URL `https://cds.cern.ch/record/2802918`

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

[9] Bothmann E, Janßen T, Knobbe M, Schmale T and Schumann S 2020 *SciPost Phys.* **8** 069

[10] Chen I K, Klimek M D and Perelstein M 2021 *SciPost Phys.* **10** 023

[11] Stienen B and Verheyen R 2021 *SciPost Phys.* **10** 038

[12] Pina-Otey S, Gaitan V, Sánchez F and Lux T 2020 *Phys. Rev. D* **102** 013003

[13] Klimek M D and Perelstein M 2020 *SciPost Phys.* **9** 053

[14] Gao C, Höche S, Isaacson J, Krause C and Schulz H 2020 *Phys. Rev. D* **101** 076002

[15] Heimel T, Winterhalder R, Butter A, Isaacson J, Krause C, Maltoni F, Mattelaer O and Plehn T 2023 *SciPost Phys.* **15** 141

[16] Maître D and Truong H 2021 *JHEP* **11** 066

[17] Danziger K, Janßen T, Schumann S and Siegert F 2022 *SciPost Phys.* **12** 164

[18] Catani S and Seymour M H 1997 *Nucl. Phys. B* **485** 291–419 [Erratum: Nucl.Phys.B 510, 503–504 (1998)]

[19] Janßen T, Maître D, Schumann S, Siegert F and Truong H 2023 *SciPost Phys.* **15** 107

[20] Maître D and Truong H 2023 *Journal of High Energy Physics* **5** 159

[21] Abadi M *et al.* 2015 TensorFlow: Large-scale machine learning on heterogeneous systems software available from `tensorflow.org`

[22] Chollet F *et al.* 2015 Keras `https://keras.io`

[23] Bai J, Lu F, Zhang K *et al.* 2019 Onnx: Open neural network exchange `https://github.com/onnx/onnx`

[24] ONNX Runtime developers 2018 Onnx runtime `https://github.com/microsoft/onnxruntime`

[25] Krauss F, Kuhn R and Soff G 2002 *JHEP* **02** 044

[26] Bothmann E, Giele W, Hoeche S, Isaacson J and Knobbe M 2022 *SciPost Phys. Codebases* 3

[27] Gleisberg T and Höche S 2008 *JHEP* **12** 039