FERMILAB-PUB-23-032-T
MCnet-23-02

Efficient phase-space generation for hadron collider event simulation

Enrico Bothmann,[1] Taylor Childers,[2] Walter Giele,[3] Florian Herren,[3]
Stefan Höche,[3] Joshua Isaacson,[3] Max Knobbe,[1] and Rui Wang[2]
[1]Institut für Theoretische Physik, Georg-August-Universität Göttingen, 37077 Göttingen, Germany
[2]Argonne National Laboratory, Lemont, IL, 60439, USA
[3]Fermi National Accelerator Laboratory, Batavia, IL 60510, USA

We present a simple yet efficient algorithm for phase-space integration at hadron colliders. Individual mappings consist of a single t-channel combined with any number of s-channel decays, and are constructed using diagrammatic information. The factorial growth in the number of channels is tamed by providing an option to limit the number of s-channel topologies. We provide a publicly available, parallelized code in C++ and test its performance in typical LHC scenarios.

I. INTRODUCTION

The problem of phase-space integration is omnipresent in particle physics. Efficient methods to evaluate phase-space integrals are needed in order to predict cross sections and decay rates for a variety of experiments, and they are required for both theoretical calculations and event simulation. In many cases, the integrand to be evaluated features a number of narrow peaks, corresponding to the resonant production of unstable massive particles. In other cases, the integrand has intricate discontinuities, arising from cuts to avoid the singular regions of scattering matrix elements in theories with massless force carriers, such as QED and QCD. In most interesting scenarios, the phase space is high dimensional, such that analytic integration is ruled out, and Monte-Carlo (MC) integration becomes the only viable option.

Many techniques have been devised to deal with this problem [1–10]. Among the most successful ones are factorization based approaches [1–3] and multi-channel integration techniques [11]. They allow to map the structure of the integral to the diagrammatic structure of the integrand. For scalar theories, and ignoring the effect of phase-space cuts, this corresponds to an ideal variable transformation. Realistic multi-particle production processes are much more complex, both because of the non-scalar nature of most of the elementary particles, and because of phase-space restrictions. Adaptive Monte-Carlo methods [12–17] are therefore used by most theoretical calculations and event generators to map out structures of the integrand which are difficult to predict. More recently, neural networks have emerged as a promising tool for this particular task [18–25].

In this letter, we introduce a novel phase-space integrator which combines several desirable features of different existing approaches. In particular, we address the computational challenges discussed in a number of reports of the HEP Software Foundation [26–28] and the recent Snowmass community study [29]. Our algorithm is based on the highly successful integration techniques employed in MCFM [30–33], combined with a standard recursive approach for s-channel topologies as used in many modern simulation programs. We provide a stand-alone implementation, which we call Chili (Common High-energy Integration LIbrary)[1], which includes the Vegas algorithm [12] and MPI parallelization. We also implement Python bindings via nanobind [34] and to Tensorflow [35], providing an interface the normalizing-flow based neural network integration frameworks iFlow [20] and MadNIS [22]. To assess the performance of our new code, we combine it with the matrix-element generators in the general-purpose event generator Sherpa [8, 36] and devise a proof of concept for the computation of real-emission next-to-leading order corrections by adding a forward branching generator which makes use of the phase-space mappings of the Catani-Seymour dipole subtraction formalism [37, 38].

The outline of the paper is as follows: Section II discusses the algorithms used in our new generator. Section III presents performance measures obtained in combination with Comix [8], and Amegic [39], and Sec. IV includes a summary and outlook.

II. THE ALGORITHM

One of the most versatile approaches to phase-space integration for high-energy collider experiments is to employ the factorization properties of the n-particle phase-space integral [3]. Consider a $2 \to n$ scattering process, where we

[1] The source code can be found at `https://gitlab.com/spice-mc/chili`.

This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics.

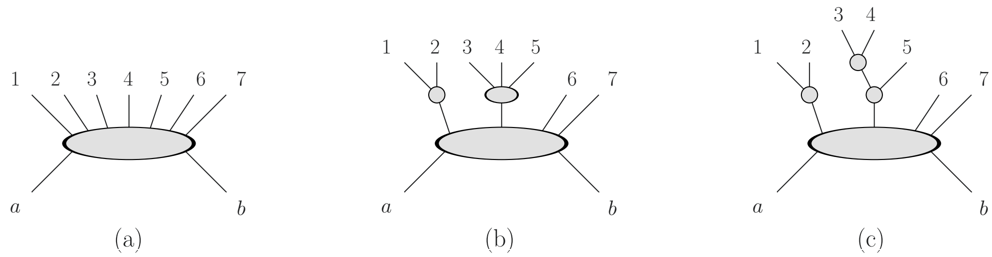

FIG. 1. Example application of the phase-space factorization formula, Eq. (2). Particles 1 through 7 are produced in the collision of particles a and b. Figure (a) represents a pure t-channel configuration, cf. Sec. II A. In Fig. (b), the differential 7-particle phase-space element is factorized into the production of four particles, two of which are the pseudo-particles $\{1,2\}$ and $\{3,4,5\}$, which subsequently decay. In Fig. (c), the decay of $\{3,4,5\}$ is again factorized into two consecutive decays.

label the incoming particles by a and b and outgoing particles by $1\ldots n$. The corresponding n-particle differential phase-space element reads

$$\mathrm{d}\Phi_n(a,b;1,\ldots,n) = \left[\prod_{i=1}^{n} \frac{\mathrm{d}^3\vec{p}_i}{(2\pi)^3\,2E_i}\right]\,(2\pi)^4\delta^{(4)}\left(p_a+p_b-\sum_{i=1}^{n}p_i\right)\,. \tag{1}$$

Following Ref. [1], the full differential phase-space element can be reduced to lower-multiplicity differential phase-space elements as follows:

$$\mathrm{d}\Phi_n(a,b;1,\ldots,n) = \mathrm{d}\Phi_{n-m+1}(a,b;\pi,m+1,\ldots,n)\,\frac{\mathrm{d}s_\pi}{2\pi}\,\mathrm{d}\Phi_m(\pi;1,\ldots,m)\,, \tag{2}$$

where π indicates an intermediate pseudo-particle of virtuality $s_\pi = p_\pi^2$. Equation (2) allows to compose the full differential phase-space element from building blocks which correspond to a single t-channel production process and a number of s-channel decays, as depicted in Fig. 1. By repeated application of Eq. (2), all decays can be reduced to two-particle decays, with differential phase-space elements $\mathrm{d}\Phi_2$. This allows to match the structure of the phase-space integral onto the structure of the Feynman diagrams in the integrand at hand, a technique that is known as diagram-based integration.

A. The t- and s-channel building blocks

In this subsection, we first describe the techniques to perform the integration using a pure t-channel differential phase-space element, $\mathrm{d}\Phi_n(a,b;1,\ldots,n)$. The final-state momenta p_1 through p_n can be associated with on-shell particles, or they can correspond to intermediate pseudo-particles whose virtuality is an additional integration variable. We start with the single-particle differential phase-space element in Eq. (1). It can be written in the form

$$\frac{\mathrm{d}^3\vec{p}_i}{(2\pi)^3\,2E_i} = \frac{1}{16\pi^2}\,\mathrm{d}p_{i,\perp}^2\,\mathrm{d}y_i\,\frac{\mathrm{d}\phi_\mathrm{i}}{2\pi}\,, \tag{3}$$

where $p_{i,\perp}$, y_i and ϕ_i are the transverse momentum, rapidity and azimuthal angle of momentum i in the laboratory frame, respectively. Many experimental analyses at hadron colliders require cuts on the transverse momentum and rapidity of jets and other analysis objects, which are easily implemented in this parametrization, leading to an excellent efficiency of the integration algorithm.

The remaining task is to implement the delta function in Eq. (1). This is achieved by combining the integral over one of the momenta, say p_n, with the integration over the light-cone momentum fractions used to convolute the partonic cross section with the PDFs. We obtain

$$\begin{aligned}\mathrm{d}x_a\mathrm{d}x_b\,\mathrm{d}\Phi_n(a,b;1,\ldots,n) = \frac{\mathrm{d}P_+\mathrm{d}P_-}{s}\,&\left[\prod_{i=1}^{n-1}\frac{1}{16\pi^2}\,\mathrm{d}p_{i,\perp}^2\,\mathrm{d}y_i\,\frac{\mathrm{d}\phi_\mathrm{i}}{2\pi}\right]\\ &\times\frac{\mathrm{d}^4p_n}{(2\pi)^3}\,\delta(p_n-s_n)\Theta(E_n)\,(2\pi)^4\delta^{(4)}\left(p_a+p_b-\sum_{i=1}^{n-1}p_i-p_n\right)\,,\end{aligned} \tag{4}$$

where s is the hadronic center-of-mass energy, and $P_\pm = P_0 \pm P_z$ is defined using $P = \sum_{i=1}^{n-1} p_i$. Changing the integration variables from P_+ and P_- to s_n and y_n, it is straightforward to evaluate the delta functions, and we obtain the final expression

$$\mathrm{d}x_a \mathrm{d}x_b\, \mathrm{d}\Phi_n(a,b;1,\dots,n) = \frac{2\pi}{s}\left[\prod_{i=1}^{n-1} \frac{1}{16\pi^2}\, \mathrm{d}p_{i,\perp}^2\, \mathrm{d}y_i\, \frac{\mathrm{d}\phi_\mathrm{i}}{2\pi}\right] \mathrm{d}y_n\ . \tag{5}$$

This form of the differential phase-space element is particularly suited for the production of electroweak vector bosons (W, Z and γ) in association with any number of jets. However, it may not be optimal for phase-space generation when there are strong hierarchies in transverse momenta of the jets, that may be better described by phase-space mappings similar to Fig. 1 (c).

The differential decay phase-space elements occurring in Fig. 1 (b) and (c) are easily composed from the corresponding expressions for two-body decays. In the frame of a time-like momentum P, this differential phase-space element can be written as

$$\mathrm{d}\Phi_2(\{1,2\};1,2) = \frac{1}{16\pi^2}\frac{\sqrt{(p_1P)^2 - p_1^2P^2}}{(p_1+p_2)P}\, \mathrm{d}\cos\theta_1^{(P)} \mathrm{d}\phi_1^{(P)}\ . \tag{6}$$

Typically, this is evaluated in the center-of-mass frame of the combined momentum, $p_1 + p_2$, where it simplifies to

$$\mathrm{d}\Phi_2(\{1,2\};1,2) = \frac{1}{16\pi^2}\frac{\sqrt{(p_1p_2)^2 - p_1^2p_2^2}}{(p_1+p_2)^2}\, \mathrm{d}\cos\theta_1^{\{1,2\}}\, \mathrm{d}\phi_1^{\{1,2\}}\ . \tag{7}$$

Equations (5) and (7) form the basic building blocks of our algorithm.

B. The multi-channel

An optimal integrator for a particular squared Feynman diagram would be composed of a combination of the t-channel map in Eq. (5) and potentially a number of s-channel maps in Eq. (7), as sketched for various configurations in Fig. 1. The complete integrand will almost never consist of a single Feynman diagram squared, and it is therefore more appropriate to combine various such integrators in order to map out different structures in the full integrand.[2] Each of those mappings is conventionally called a phase-space "channel", and each channel is a valid phase-space integrator in it's own right. They can be combined using the multi-channel technique, which was introduced in [11]. We refer the reader to the original publication for the details of this method. Here we will briefly describe how the individual channels are constructed in our integrator.

We begin by extracting the three-particle vertices from the interaction model. Given a set of external flavors, we can use the vertex information to construct all possible topologies of Feynman diagrams with the maximum number of propagators. For each topology, we apply the following algorithm: If an s-channel propagator is found, we use the factorization formula, Eq. (2) to split the differential phase-space element into a production and a decay part. This procedure starts with the external states and it is repeated until no more factorization is possible. As the number of possible s-channel topologies grows factorially in many cases, our algorithm provides an option to limit the maximum number of s-channels that are implemented. This helps to tailor the integrator to the problem at hand and allows to control the computational complexity.

Following standard practice, we generate the virtuality of the intermediate s-channel pseudo-particles using a Breit-Wigner distribution if the particle has a mass and width, or following a $\mathrm{d}s/s^\alpha$ distribution ($\alpha < 1$), if the particle is massless. The transverse momenta in Eq. (5) are generated according to $\mathrm{d}p_\perp^2/(2p_{\perp,c} + p_\perp)^2$, where $p_{\perp,c}$ is an adjustable parameter that can be used to maximize efficiency, e.g. by setting it to the jet transverse momentum cut. The rapidities in Eq. (5) and the angles in Eq. (7) are generated using a flat prior distribution.

C. Next-to-leading order calculations and dipole mappings

The integration of real-emission corrections in next-to-leading order QCD or QED calculations poses additional challenges for a phase-space integration algorithm. In order to achieve a local cancellation of singularities, subtraction

[2] An alternative option is to partition the integrand into terms which exhibit the structure of an individual diagram [6].

methods are typically employed in these calculations [37, 40]. This makes the behavior of the integrand less predictable than at leading order, and therefore complicates the construction of integration channels. Various approaches have been devised to deal with the problem. We adopt a solution that is based on the on-shell momentum mapping technique used in the Catani-Seymour dipole subtraction scheme [37, 38] and that has long been used in generators such as MCFM [32, 33, 41] and MUNICH [42].[3]

Following Ref. [37], there are four different types of local infrared subtraction terms that are used to make real-emission corrections and virtual corrections in NLO calculations separately infrared finite. They are classified according to the type of collinear divergence (initial state or final state) and the type of color spectator parton (initial state or final state). The massless on-shell phase-space mapping for the final-final configuration (FF) reads

$$\mathrm{d}\Phi_n^{(\mathrm{FF})}(a,b;1,\ldots,n) = \mathrm{d}\Phi_{n-1}(a,b;1,\ldots,\widetilde{\imath\jmath},\ldots,\tilde{k},\ldots,n)\,\frac{2\tilde{p}_{ij}\tilde{p}_k}{16\pi^2}\,\mathrm{d}y_{ij,k}\mathrm{d}\tilde{z}_i\,\frac{\mathrm{d}\phi}{2\pi}\,(1-y_{ij,k})\;. \tag{8}$$

where

$$p_i^\mu = \tilde{z}_i\,\tilde{p}_{ij}^\mu + (1-\tilde{z}_i)\,y_{ij,k}\,\tilde{p}_k^\mu + k_\perp^\mu\;,\qquad p_k^\mu = (1-y_{ij,k})\,\tilde{p}_k^\mu\;,\qquad p_j^\mu = \tilde{p}_{ij} + \tilde{p}_k - p_i - p_k\;, \tag{9}$$

and where $k_\perp^2 = -\tilde{z}_i(1-\tilde{z}_i)y_{ij,k}\,2\tilde{p}_{ij}\tilde{p}_k$ is determined by the on-shell conditions.
The massless on-shell phase-space mapping for the final-initial and initial-final configurations (FI/IF) reads

$$\mathrm{d}\Phi_n^{(\mathrm{FI/IF})}(a,b;1,\ldots,n) = \mathrm{d}\Phi_{n-1}(\tilde{a},b;1,\ldots,\widetilde{\imath\jmath},\ldots,n)\,\frac{2\tilde{p}_{ij}p_a}{16\pi^2}\,\mathrm{d}\tilde{z}_i\mathrm{d}x_{ij,a}\,\frac{\mathrm{d}\phi}{2\pi}\;. \tag{10}$$

where

$$p_i^\mu = \tilde{z}_i\,\tilde{p}_{ij}^\mu + (1-\tilde{z}_i)\,\frac{1-x_{ij,a}}{x_{ij,a}}\,\tilde{p}_a^\mu + k_\perp^\mu\;,\qquad p_a^\mu = \frac{1}{x_{ij,a}}\,\tilde{p}_a^\mu\;,\qquad p_j^\mu = \tilde{p}_{ij} - \tilde{p}_a + \tilde{p}_a - \tilde{p}_i\;, \tag{11}$$

and where $k_\perp^2 = -\tilde{z}_i(1-\tilde{z}_i)(1-x_{ij,a})/x_{ij,a}\,2\tilde{p}_{ij}\tilde{p}_a$.
The massless on-shell phase-space mapping for the initial-initial configurations (II) reads

$$\mathrm{d}\Phi_n^{(\mathrm{II})}(a,b;1,\ldots,n) = \mathrm{d}\Phi_{n-1}(\widetilde{ai},b;\tilde{1},\ldots,\tilde{n})\,\frac{2p_ap_b}{16\pi^2}\,\mathrm{d}\tilde{v}_i\mathrm{d}x_{i,ab}\,\frac{\mathrm{d}\phi}{2\pi}\;. \tag{12}$$

where

$$p_i^\mu = \frac{1-x_{i,ab}-\tilde{v}_i}{x_{i,ab}}\,\tilde{p}_a^\mu + \tilde{v}_i\,p_b^\mu + k_\perp^\mu\;,\qquad p_a^\mu = \frac{1}{x_{i,ab}}\,\tilde{p}_{ai}^\mu\;,\qquad p_j^\mu = \Lambda^\mu{}_\nu(K,\tilde{K})\tilde{p}_j^\nu\quad\forall j\in\{1,\ldots,n\}, j\neq i\;, \tag{13}$$

and where $k_\perp^2 = -(1-x_{i,ab}-\tilde{v})/x_{ij,a}\,\tilde{v}_i\,2\tilde{p}_{ai}p_b$. The transformation, $\Lambda^\mu{}_\nu(K,\tilde{K})$, is defined in Sec. 5.5 of Ref. [37]. The three above mappings are sufficient to treat any real-emission correction in massless QCD. We infer the possible dipole configurations from the flavor structure of the process and combine all possible mappings into a multi-channel integrator [11].

D. Combination with normalizing-flow based integrators

With the development of modern machine learning methods, new techniques for adaptive Monte-Carlo integration have emerged, which are based on the extension [43, 44] of a nonlinear independent components estimation technique [45, 46], also known as a normalizing flow. They have been used to develop integration algorithms based on existing multi-channel approaches [19, 21, 22, 25]. One of the main obstacles to scaling such approaches to high multiplicity has been the fact that the underlying phase-space mappings are indeed multi channels, which induces hyperparameters that increase the dimensionality of the optimization problem. Here we propose a different strategy. We observe that the basic t-channel integration algorithm implementing Eq. (5) requires the minimal amount of random numbers, and shows a good efficiency (cf. Sec. III). It is therefore ideally suited to provide a basic mapping of the n-particle phase space at hadron colliders into a $3n-4+2$ dimensional unit hypercube, required for combination with normalizing-flow based integrators. We provide Python bindings in Chili via nanobind [34] and a dedicated Tensorflow [35] interface. This allows the use of the iFlow [20] and MadNIS [22] frameworks to test this idea, and to evaluate the performance of this novel algorithm.

---

[3] We make this feature available only for use within Sherpa, but a future version of our stand-alone code will support it as well.

Process	Sherpa		Chili		Chili (basic)	
	$\Delta\sigma/\sigma$ 6M pts	η 100 evts	$\Delta\sigma/\sigma$ 6M pts	η 100 evts	$\Delta\sigma/\sigma$ 6M pts	η 100 evts
W^++1j	0.5‰	7×10^{-2}	0.6‰	9×10^{-2}	0.6‰	9×10^{-2}
W^++2j	1.2‰	9×10^{-3}	1.1‰	2×10^{-2}	1.2‰	1×10^{-2}
W^++3j	2.0‰	1×10^{-3}	2.0‰	4×10^{-3}	2.9‰	2×10^{-3}
W^++4j	3.7‰	2×10^{-4}	4.9‰	7×10^{-4}	6.0‰	3×10^{-4}
W^++5j	7.2‰	4×10^{-5}	22‰	1×10^{-5}	26‰	1×10^{-5}

Process	Sherpa		Chili		Chili (basic)	
	$\Delta\sigma/\sigma$ 6M pts	η 100 evts	$\Delta\sigma/\sigma$ 6M pts	η 100 evts	$\Delta\sigma/\sigma$ 6M pts	η 100 evts
Z+1j	0.4‰	2×10^{-1}	0.5‰	1×10^{-1}	0.5‰	1×10^{-1}
Z+2j	0.8‰	2×10^{-2}	0.8‰	3×10^{-2}	1.0‰	2×10^{-2}
Z+3j	1.3‰	4×10^{-3}	1.6‰	7×10^{-3}	2.5‰	4×10^{-3}
Z+4j	2.2‰	8×10^{-4}	3.6‰	1×10^{-3}	5.0‰	6×10^{-4}
Z+5j	3.7‰	1×10^{-4}	11‰	1×10^{-4}	13‰	2×10^{-4}

Process	Sherpa		Chili		Chili (basic)	
	$\Delta\sigma/\sigma$ 6M pts	η 100 evts	$\Delta\sigma/\sigma$ 6M pts	η 100 evts	$\Delta\sigma/\sigma$ 6M pts	η 100 evts
h+1j	0.4‰	2×10^{-1}	0.4‰	2×10^{-1}	0.4‰	2×10^{-1}
h+2j	0.8‰	2×10^{-2}	0.6‰	5×10^{-2}	0.6‰	5×10^{-2}
h+3j	1.4‰	3×10^{-3}	0.9‰	2×10^{-2}	0.9‰	2×10^{-2}
h+4j	2.4‰	6×10^{-4}	1.6‰	6×10^{-3}	1.7‰	7×10^{-3}
h+5j	4.5‰	1×10^{-4}	3.2‰	1×10^{-3}	3.6‰	1×10^{-3}

Process	Sherpa		Chili		Chili (basic)	
	$\Delta\sigma/\sigma$ 6M pts	η 100 evts	$\Delta\sigma/\sigma$ 6M pts	η 100 evts	$\Delta\sigma/\sigma$ 6M pts	η 100 evts
$t\bar{t}$+0j	0.6‰	1×10^{-1}	0.6‰	1×10^{-1}	0.6‰	1×10^{-1}
$t\bar{t}$+1j	0.9‰	2×10^{-2}	0.6‰	6×10^{-2}	0.9‰	3×10^{-2}
$t\bar{t}$+2j	1.4‰	4×10^{-3}	0.9‰	2×10^{-2}	1.4‰	1×10^{-2}
$t\bar{t}$+3j	2.6‰	7×10^{-4}	1.5‰	7×10^{-3}	2.9‰	2×10^{-3}
$t\bar{t}$+4j	4.0‰	1×10^{-4}	3.2‰	1×10^{-3}	3.5‰	8×10^{-4}

Process	Sherpa		Chili		Chili (basic)	
	$\Delta\sigma/\sigma$ 6M pts	η 100 evts	$\Delta\sigma/\sigma$ 6M pts	η 100 evts	$\Delta\sigma/\sigma$ 6M pts	η 100 evts
γ+1j	0.4‰	2×10^{-1}	0.6‰	1×10^{-1}	0.6‰	1×10^{-1}
γ+2j	1.1‰	7×10^{-3}	2.2‰	3×10^{-3}	3.7‰	1×10^{-3}
γ+3j	2.4‰	5×10^{-4}	4.9‰	4×10^{-4}	10‰	1×10^{-4}
γ+4j	5.0‰	7×10^{-5}	20‰	3×10^{-5}	30‰	4×10^{-5}
γ+5j	9.3‰	2×10^{-5}	28‰	7×10^{-6}	36‰	2×10^{-6}

Process	Sherpa		Chili		Chili (basic)	
	$\Delta\sigma/\sigma$ 6M pts	η 100 evts	$\Delta\sigma/\sigma$ 6M pts	η 100 evts	$\Delta\sigma/\sigma$ 6M pts	η 100 evts
2jets	0.6‰	5×10^{-2}	0.4‰	1×10^{-1}	0.5‰	7×10^{-2}
3jets	1.2‰	5×10^{-3}	1.0‰	1×10^{-2}	1.8‰	7×10^{-3}
4jets	2.5‰	5×10^{-4}	2.0‰	3×10^{-3}	3.4‰	1×10^{-3}
5jets	4.7‰	9×10^{-5}	5.1‰	6×10^{-4}	8.1‰	2×10^{-4}
6jets	7.0‰	2×10^{-5}	15‰	5×10^{-5}	14‰	4×10^{-5}

TABLE I. Relative Monte-Carlo uncertainties, $\Delta\sigma/\sigma$, and unweighting efficiencies, η, in leading-order calculations. The center-of-mass energy is $\sqrt{s} = 14$ TeV, jets are defined using the anti-k_T algorithm with $p_{\perp,j} = 30$ GeV and $|y_j| \leq 6$. Vegas grids and multi-channel weights have been adapted using 1.2M non-zero phase-space points. For details see the main text.

III. PERFORMANCE BENCHMARKS

In this section we present first numerical results obtained with our new integrator, Chili. We have interfaced the new framework with the general-purpose event generator Sherpa [36, 47, 48], which is used to compute the partonic matrix elements and the parton luminosity with the help of Comix [8] and Amegic [39]. To allow performance tests from low to high particle multiplicity, we perform color sampling. This affects the convergence rate, and we note that better MC uncertainties could in principle be obtained for color-summed computations, but at the cost of much larger computing time at high multiplicity. The performance comparison between Sherpa and Chili would, however, be unaffected. We use the NNPDF 3.0 PDF set [49] at NNLO precision, and the corresponding settings of the strong coupling, i.e. $\alpha_s(m_z) = 0.118$ and running to 3-loop order. Light quarks, charm and bottom quarks are assumed to be massless, and we set $m_t = 173.21$. The electroweak parameters are determined in the complex mass scheme using the inputs $\alpha(m_Z) = 1/128.8$, $m_W = 80.385$, $m_Z = 91.1876$, $m_h = 125$ and $\Gamma_W = 2.085$, $\Gamma_Z = 2.4952$. We assume incoming proton beams at a hadronic center-of-mass energy of $\sqrt{s} = 14$ TeV. To implement basic phase-space cuts, we reconstruct jets using the anti-k_T jet algorithm [50] with $R = 0.4$ in the implementation of FastJet [51] and require $p_{\perp,j} \geq 30$ GeV and $|y_j| \leq 6$. Photons are isolated from QCD activity based on Ref. [52] with δ_0=0.4, n=2 and ϵ_γ=2.5% and are required to have $p_{\perp,\gamma} \geq 30$ GeV. All results presented in this section are obtained with a scalable version of our new integrator using parallel execution on CPUs with the help of MPI.

Table I shows a comparison between MC uncertainties and event generation efficiencies in leading-order calculations, obtained with the recursive phase-space generator in Comix and with Chili. To improve the convergence of the integrals we use the Vegas [12] algorithm, which is implemented independently in both Sherpa and Chili. The MC uncertainties are given after optimizing the adaptive integrator with 1.2 million non-zero phase-space points and evaluation of the integral with 6 million non-zero phase-space points. We employ the definition of event generation efficiency in Ref. [21], and we evaluate it using 100 replicas of datasets leading to 100 unweighted events each. We test the production of W^+ and Z bosons with leptonic decay, on-shell Higgs boson production, top-quark pair production, direct photon production and pure jet production. These processes are omnipresent in background simulations at the Large Hadron Collider (LHC), and are typically associated with additional light jet activity due to the large phase space. Accordingly, we test the basic process with up to four additional light jets. In single boson production we do not include the trivial process without any light jets. We observe that the performance of our new integrator is well comparable and in many cases slightly better than the performance of the recursive phase-space generator

Process	Sherpa		Chili		Chili (basic)	
	$\Delta\sigma/\sigma$	η	$\Delta\sigma/\sigma$	η	$\Delta\sigma/\sigma$	η
boosted	6M pts	100 evts	6M pts	100 evts	6M pts	100 evts
W^++2j	1.4‰	4×10^{-3}	1.4‰	8×10^{-3}	2.5‰	2×10^{-3}
W^++3j	2.5‰	9×10^{-4}	3.8‰	6×10^{-4}	6.9‰	2×10^{-4}
W^++4j	4.2‰	2×10^{-4}	10‰	7×10^{-5}	17‰	4×10^{-5}
W^++5j	7.2‰	4×10^{-5}	27‰	3×10^{-6}	48‰	4×10^{-6}

Process	Sherpa		Chili		Chili (basic)	
	$\Delta\sigma/\sigma$	η	$\Delta\sigma/\sigma$	η	$\Delta\sigma/\sigma$	η
boosted	6M pts	100 evts	6M pts	100 evts	6M pts	100 evts
Z+2j	1.0‰	9×10^{-3}	1.1‰	1×10^{-2}	1.8‰	6×10^{-3}
Z+3j	1.6‰	2×10^{-3}	2.5‰	2×10^{-3}	5.0‰	5×10^{-4}
Z+4j	2.8‰	4×10^{-4}	7.6‰	2×10^{-4}	27‰	6×10^{-5}
Z+5j	4.6‰	9×10^{-5}	15‰	3×10^{-5}	33‰	2×10^{-5}

Process	Sherpa		Chili		Chili (basic)	
	$\Delta\sigma/\sigma$	η	$\Delta\sigma/\sigma$	η	$\Delta\sigma/\sigma$	η
boosted	6M pts	100 evts	6M pts	100 evts	6M pts	100 evts
h+2j	1.1‰	8×10^{-3}	0.7‰	4×10^{-2}	0.7‰	3×10^{-2}
h+3j	1.8‰	2×10^{-3}	1.0‰	1×10^{-2}	1.1‰	1×10^{-2}
h+4j	3.0‰	4×10^{-4}	1.7‰	3×10^{-3}	1.6‰	4×10^{-3}
h+5j	4.8‰	9×10^{-5}	4.2‰	7×10^{-4}	3.1‰	1×10^{-4}

Process	Sherpa		Chili		Chili (basic)	
	$\Delta\sigma/\sigma$	η	$\Delta\sigma/\sigma$	η	$\Delta\sigma/\sigma$	η
boosted	6M pts	100 evts	6M pts	100 evts	6M pts	100 evts
γ+2j	1.4‰	4×10^{-3}	2.3‰	2×10^{-3}	2.3‰	2×10^{-3}
γ+3j	2.3‰	7×10^{-4}	4.3‰	4×10^{-4}	9.0‰	1×10^{-4}
γ+4j	4.0‰	2×10^{-4}	9.9‰	1×10^{-4}	25‰	1×10^{-5}
γ+5j	7.3‰	2×10^{-5}	36‰	1×10^{-6}	49‰	3×10^{-6}

Process	Sherpa		Chili		Chili (basic)	
	$\Delta\sigma/\sigma$	η	$\Delta\sigma/\sigma$	η	$\Delta\sigma/\sigma$	η
boosted	6M pts	100 evts	6M pts	100 evts	6M pts	100 evts
$t\bar{t}$+1j	1.0‰	1×10^{-2}	0.7‰	4×10^{-2}	1.5‰	1×10^{-2}
$t\bar{t}$+2j	2.0‰	1×10^{-3}	1.1‰	1×10^{-2}	2.3‰	2×10^{-3}
$t\bar{t}$+3j	3.2‰	4×10^{-4}	1.9‰	3×10^{-3}	3.7‰	8×10^{-4}
$t\bar{t}$+4j	4.9‰	1×10^{-4}	3.8‰	7×10^{-4}	8.4‰	2×10^{-4}

Process	Sherpa		Chili		Chili (basic)	
	$\Delta\sigma/\sigma$	η	$\Delta\sigma/\sigma$	η	$\Delta\sigma/\sigma$	η
m_{jj} cut	6M pts	100 evts	6M pts	100 evts	6M pts	100 evts
h+2j	0.9‰	1×10^{-2}	0.8‰	1×10^{-2}	0.9‰	1×10^{-2}
h+3j	1.9‰	1×10^{-3}	1.2‰	5×10^{-3}	1.3‰	4×10^{-3}
h+4j	4.1‰	2×10^{-4}	1.8‰	2×10^{-3}	2.3‰	1×10^{-3}
h+5j	16‰	5×10^{-5}	5.0‰	2×10^{-4}	4.5‰	5×10^{-4}

TABLE II. Relative Monte-Carlo uncertainties, $\Delta\sigma/\sigma$, and unweighting efficiencies, η, in leading-order calculations for boosted event topologies. The center-of-mass energy is $\sqrt{s} = 14$ TeV, jets are defined using the anti-k_T algorithm with $p_{\perp,j} = 30$ GeV and $|y_j| \leq 6$. We require a leading jet at $p_{\perp,j1} \geq 300$ GeV. Vegas grids and multi-channel weights have been adapted using 1.2M non-zero phase-space points. For details see the main text.

Process 1M pts	Sherpa $\Delta\sigma/\sigma$	Sherpa $\varepsilon_{\rm cut}$	Chili (basic) $\Delta\sigma/\sigma$	Chili (basic) $\varepsilon_{\rm cut}$
W^++1j / B-like	1.3‰	43%	1.4‰	99%
R-like	4.1‰	46%	3.6‰	58%
W^++2j / B-like	2.2‰	37%	4.4‰	99%
R-like	1.4%	74%	1.5%	80%
W^++3j†/ B-like	2.8%	33%	3.5%	97%
R-like	3.0%	75%	4.3%	87%

Process 1M pts	Sherpa $\Delta\sigma/\sigma$	Sherpa $\varepsilon_{\rm cut}$	Chili (basic) $\Delta\sigma/\sigma$	Chili (basic) $\varepsilon_{\rm cut}$
h+1j / B-like	1.3‰	56%	0.7‰	99%
R-like	3.0‰	52%	2.1‰	69%
h+2j / B-like	2.6‰	34%	1.4‰	99%
R-like	8.1‰	68%	8.2‰	87%
h+3j*/ B-like	2.3%	29%	1.0%	96%
R-like	2.0%	65%	2.0%	83%

Process 1M pts	Sherpa $\Delta\sigma/\sigma$	Sherpa $\varepsilon_{\rm cut}$	Chili (basic) $\Delta\sigma/\sigma$	Chili (basic) $\varepsilon_{\rm cut}$
$t\bar{t}$+0j / B-like	0.4‰	99%	0.8‰	99%
R-like	0.2‰	99%	0.3‰	99%
$t\bar{t}$+1j / B-like	1.7‰	61%	1.7‰	99%
R-like	5.8‰	82%	5.9‰	92%
$t\bar{t}$+2j / B-like	1.5%	45%	1.0%	98%
R-like	1.4%	78%	1.7%	85%

Process 1M pts	Sherpa $\Delta\sigma/\sigma$	Sherpa $\varepsilon_{\rm cut}$	Chili (basic) $\Delta\sigma/\sigma$	Chili (basic) $\varepsilon_{\rm cut}$
2jets / B-like	1.5‰	34%	0.7‰	99%
R-like	8.3‰	76%	4.3‰	89%
3jets / B-like	4.2%	9.6%	6.1‰	88%
R-like	4.5%	56%	3.7%	81%
4jets*/ B-like	4.8%	12%	3.2%	90%
R-like	4.7%	50%	3.7%	79%

TABLE III. Relative Monte-Carlo uncertainties, $\Delta\sigma/\sigma$, and cut efficiencies, $\varepsilon_{\rm cut}$, in next-to-leading order calculations. The center-of-mass energy is $\sqrt{s} = 14$ TeV, jets are defined using the anti-k_T algorithm with $p_{\perp,j} = 30$ GeV and $|y_j| \leq 6$. The superscript † indicates a factor 10 reduction in the number of points to evaluate the Born-like components. The superscript * indicates a factor 10 reduction in the number of points to evaluate the Born-like components and the usage of a global K-factor as a stand-in for the finite virtual corrections.

in Sherpa. This is both encouraging and somewhat surprising, given the relative simplicity of our new approach, which does not make use of repeated t-channel factorization. Due to the uniform jet cuts, we even obtain similar performance when using the minimal number of s-channel parametrizations. This setup is labeled as Chili (basic) in Tab. I. The results suggest that a single phase-space parametrization may in many cases be sufficient to compute cross sections and generate events at high precision, which is advantageous in terms of computing time and helps to scale the computation to higher multiplicity processes. Moreover, it circumvents the problems related to multi-channel integration discussed in [21, 22] when combining our integrator with neural network based adaptive random number mapping techniques. We note that this configuration is also used by MCFM [30].

Table II shows a similar comparison as in Tab. I, but in addition we apply a cut on the leading jet, requiring

Process	CHILI		CHILI (Basic)+NF		Process	CHILI		CHILI (Basic)+NF	
(color sum)	$\Delta\sigma/\sigma$ 6M pts	η 100 evts	$\Delta\sigma/\sigma$ 6M pts	η 100 evts	(color sum)	$\Delta\sigma/\sigma$ 6M pts	η 100 evts	$\Delta\sigma/\sigma$ 6M pts	η 100 evts
W^++1j	0.4‰	2×10^{-1}	0.2‰	4×10^{-1}	Z+1j	0.4‰	2×10^{-1}	0.1‰	5×10^{-1}
W^++2j	0.7‰	4×10^{-2}	0.7‰	5×10^{-2}	Z+2j	0.6‰	5×10^{-2}	0.6‰	6×10^{-2}

Process	CHILI		CHILI (Basic)+NF		Process	CHILI		CHILI (Basic)+NF	
(color sum)	$\Delta\sigma/\sigma$ 6M pts	η 100 evts	$\Delta\sigma/\sigma$ 6M pts	η 100 evts	(color sum)	$\Delta\sigma/\sigma$ 6M pts	η 100 evts	$\Delta\sigma/\sigma$ 6M pts	η 100 evts
h+1j	0.2‰	5×10^{-1}	0.05‰	8×10^{-1}	$t\bar{t}$+0j	0.1‰	6×10^{-1}	0.05‰	7×10^{-1}
h+2j	0.3‰	1×10^{-1}	0.3‰	2×10^{-1}	$t\bar{t}$+1j	0.2‰	3×10^{-1}	0.3‰	2×10^{-1}

Process	CHILI		CHILI (Basic)+NF		Process	CHILI		CHILI (Basic)+NF	
(color sum)	$\Delta\sigma/\sigma$ 6M pts	η 100 evts	$\Delta\sigma/\sigma$ 6M pts	η 100 evts	(color sum)	$\Delta\sigma/\sigma$ 6M pts	η 100 evts	$\Delta\sigma/\sigma$ 6M pts	η 100 evts
γ+1j	0.6‰	2×10^{-1}	0.1‰	5×10^{-1}	2jets	0.2‰	4×10^{-1}	0.08‰	6×10^{-1}
γ+2j	1.8‰	5×10^{-3}	1.4‰	9×10^{-2}	3jets	0.5‰	6×10^{-2}	0.7‰	3×10^{-2}

TABLE IV. Relative Monte-Carlo uncertainties, $\Delta\sigma/\sigma$, and unweighting efficiencies, η, in leading-order calculations. The center-of-mass energy is $\sqrt{s} = 14$ TeV, jets are defined using the anti-k_T algorithm with $p_{\perp,j} = 30$ GeV and $|y_j| \leq 6$. For details see the main text.

$p_{\perp,j1} > 300$ GeV. This configuration tests the regime where the hard system receives a large boost, and there is usually a strong hierarchy between the jet transverse momenta. In these scenarios we expect the complete CHILI integrator to outperform the basic configuration with a t-channel only, which is confirmed by the comparison in Tab. II. The lower right sub-table shows a configuration where we do not apply the additional transverse momentum cut, but instead use a large di-jet invariant mass cut, typical for VBF searches and measurements, $m_{j1,j2} \geq 600$ GeV.

Table III shows a comparison of MC uncertainties and cut efficiencies for various next-to-leading order QCD computations. The shorthand B-like stands for the Born plus virtual plus integrated IR counterterms in the Catani-Seymour dipole subtraction method. The shorthand R-like stands for the IR-subtracted real-emission corrections using Catani-Seymour dipole subtraction. These calculations exhibit slightly different structures than at leading order in QCD, cf. [41]. The real-emission integrals in particular test the efficiency of the dipole mapping described in Sec. II C. It can be seen that our new algorithm has a much better cut efficiency than the recursive phase-space generator in SHERPA, which is again advantageous in terms of overall computing time. The MC uncertainty for a given number of phase-space points is reduced at low jet multiplicity, and generally comparable to the recursive phase-space generator. Given the simplicity of the CHILI approach, this is a very encouraging result for the development of NLO simulations on simpler computing architectures. If a speedup of the matrix-element calculation is obtained, for example through analytic expressions [53], accelerated numerical evaluation [54–57] or the usage of surrogate methods [58, 59], then the linear scaling of the basic CHILI generator at leading order, and the polynomial scaling of the dipole-based generator, will become an important feature.

Table IV shows a comparison of the Vegas-based CHILI integrator and the neural-network assisted integrator for color summed matrix elements. We use the single channel configuration of MADNIS [22] (which is consistent with iFlow [20]) in combination with CHILI. The network is setup with 6 rational quadratic spline coupling layers [60] with random permutations, each consisting of a neural network with 2 layers with 16 nodes each using a leaky ReLU activation function. The network is trained using 20 epochs of training with 100 batches of 1000 events per epoch with the variance as the loss term as in Ref. [22]. The learning rate starts at 0.001 and decays each epoch by $l_0/(1+l_d s/d_s)$, where l_0 is the initial learning rate, $l_d = 0.01$ is the decay rate, s is the number of steps, and $d_s = 100$ is the number of steps before applying the decay. Optimizing these parameters to achieve peak performance is beyond the scope of this project and can be done in a similar fashion as in Ref. [21].

Figures 2 and 3 show the weight distributions from 6 million phase-space points after training for the simplest and next to simplest of our test processes. We compare the recursive integrator of COMIX, CHILI with Vegas and CHILI in combination with MADNIS. All results have been computed using color summed matrix elements. It can be seen that the normalizing flow based integrator yields a very narrow weight distribution in most cases, leading to the excellent unweighting efficiencies shown in Tab. IV. However, the default COMIX integrator leads to a sharper upper edge of the weight distribution in the more complex scenarios of Fig. 3, which is favorable for unweighting. This indicates that the multi-channel approach with additional s-channels is favorable at high multiplicities. We will investigate further this effect using the technology developed in Ref. [22]. Furthermore, while the variance loss is optimal for achieving a narrow weight distribution, larger weight terms are not significantly penalized. This in turn leads to a less sharp upper edge in the weight distribution. Additionally, the number of points required to reach optimal performance for the normalizing flow is significantly higher than the Vegas based approaches, as demonstrated in Ref. [20]. A study of

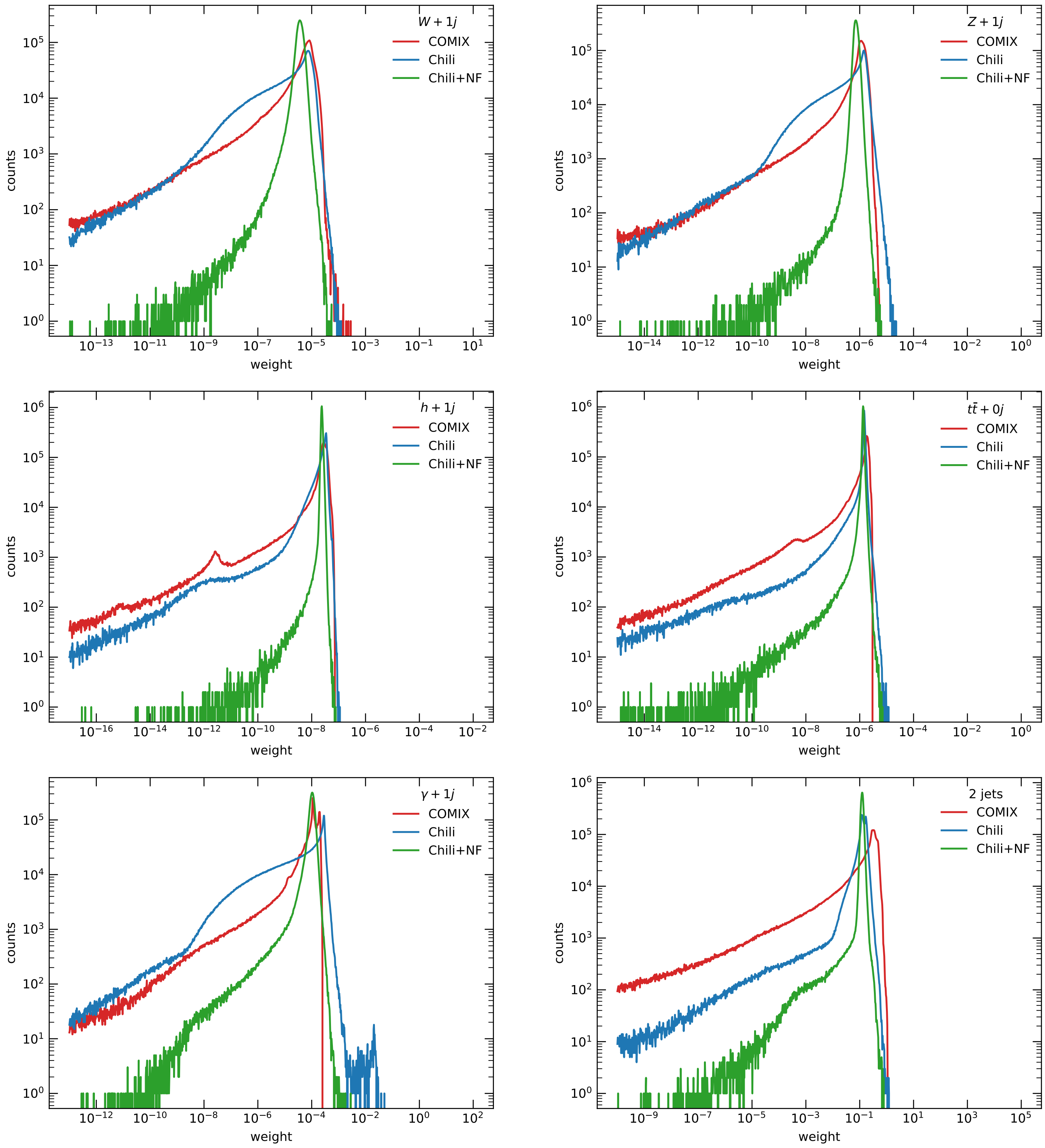

FIG. 2. Weight distribution for the lowest multiplicity processes found in Tab. IV. Each curve contains 6 million events. The Comix integrator is shown in red, the Chili with Vegas is shown in blue, and Chili with normalizing flows is shown in green. The results for $W + 1j$ is in the upper right, $Z + 1j$ in the upper left, the middle row consists of $h + 1j$ and $t\bar{t} + 0j$, and the bottom row has $\gamma + 1j$ and dijets respectively.

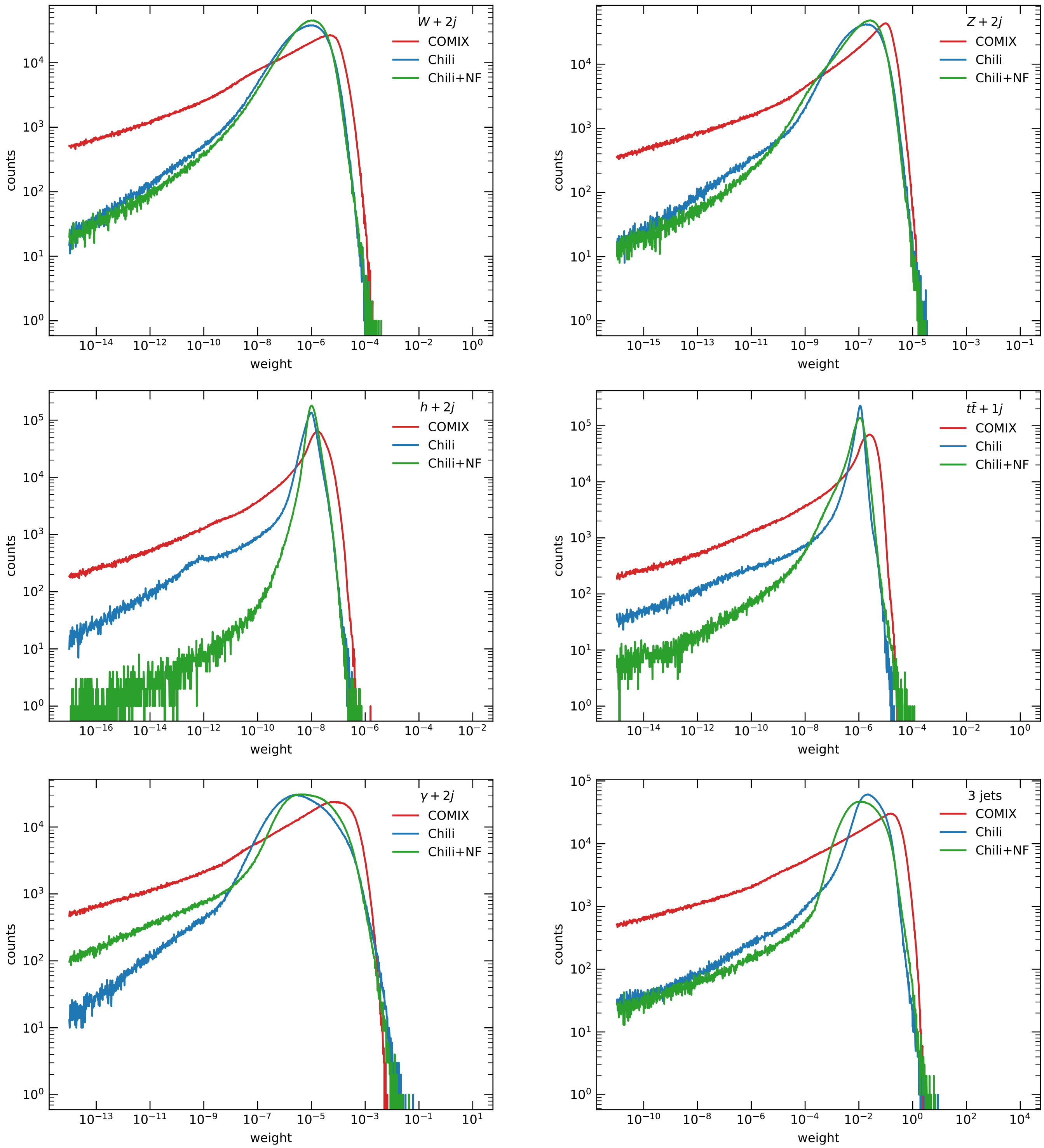

FIG. 3. Same as Fig. 2, but with an additional jet for each process.

the effect on the choice of loss function and other hyper-parameters involved in the normalizing flow approach is left to a future work to improve the unweighting efficiency at higher multiplicities and the convergence of the integrator.

IV. OUTLOOK

We have presented a new phase-space generator that combines various existing techniques for hadron collider phase-space integration into a simple and efficient algorithm. We have implemented these techniques in a scalable framework for CPU computing. Several extensions of this framework are in order: It should be ported to allow the usage of GPUs. Computing platforms other than CPUs and GPUs could be enabled with the help of Kokkos [61] or similar computing models. This becomes particularly relevant in light of recent advances in computing matrix elements on GPUs using portable programming models [54–57]. In addition, the techniques for real-emission corrections should be extended beyond SHERPA, in order to make our generator applicable to a wider range of problems. Lastly, we plan to further explore the combination of our new techniques with existing neural-network based integration methods.

ACKNOWLEDGMENTS

We thank John Campbell for many stimulating discussions and his support of the project. This research was supported by the Fermi National Accelerator Laboratory (Fermilab), a U.S. Department of Energy, Office of Science, HEP User Facility. Fermilab is managed by Fermi Research Alliance, LLC (FRA), acting under Contract No. DE–AC02–07CH11359. The work of F.H., S.H. and J.I. was supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Scientific Discovery through Advanced Computing (SciDAC) program, grant "HPC framework for event generation at colliders". F.H. acknowledges support by the Alexander von Humboldt foundation. E.B. and M.K. acknowledge support from BMBF (contract 05H21MGCAB). Their research is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 456104544; 510810461.

[1] F. James, CERN-68-15.
[2] E. Byckling and K. Kajantie, Phys. Rev. **187**, 2008 (1969).
[3] E. Byckling and K. Kajantie, Nucl. Phys. **B9**, 568 (1969).
[4] R. Kleiss, W. J. Stirling, and S. D. Ellis, Comput. Phys. Commun. **40**, 359 (1986).
[5] A. Kanaki and C. G. Papadopoulos, Comput. Phys. Commun. **132**, 306 (2000), hep-ph/0002082.
[6] F. Maltoni and T. Stelzer, JHEP **02**, 027 (2003), hep-ph/0208156.
[7] A. van Hameren and C. G. Papadopoulos, Eur. Phys. J. **C25**, 563 (2002), hep-ph/0204055.
[8] T. Gleisberg and S. Höche, JHEP **12**, 039 (2008), arXiv:0808.3674 [hep-ph].
[9] A. van Hameren, (2010), arXiv:1003.4953 [hep-ph].
[10] S. Plätzer, (2013), arXiv:1308.2922 [hep-ph].
[11] R. Kleiss and R. Pittau, Comput. Phys. Commun. **83**, 141 (1994), hep-ph/9405257.
[12] G. P. Lepage, J. Comput. Phys. **27**, 192 (1978).
[13] T. Ohl, Comput. Phys. Commun. **120**, 13 (1999), hep-ph/9806432.
[14] G. P. Lepage, J. Comput. Phys. **439**, 110386 (2021), arXiv:2009.05112 [physics.comp-ph].
[15] S. Jadach, Comput. Phys. Commun. **152**, 55 (2003), physics/0203033.
[16] T. Hahn, Comput. Phys. Commun. **168**, 78 (2005), hep-ph/0404043.
[17] A. van Hameren, Acta Phys. Polon. B **40**, 259 (2009), arXiv:0710.2448 [hep-ph].
[18] M. D. Klimek and M. Perelstein, SciPost Phys. **9**, 053 (2020), arXiv:1810.11509 [hep-ph].
[19] E. Bothmann, T. Janßen, M. Knobbe, T. Schmale, and S. Schumann, SciPost Phys. **8**, 069 (2020), arXiv:2001.05478 [hep-ph].
[20] C. Gao, J. Isaacson, and C. Krause, Mach. Learn. Sci. Tech. **1**, 045023 (2020), arXiv:2001.05486 [physics.comp-ph].
[21] C. Gao, S. Höche, J. Isaacson, C. Krause, and H. Schulz, Phys. Rev. D **101**, 076002 (2020), arXiv:2001.10028 [hep-ph].
[22] T. Heimel, R. Winterhalder, A. Butter, J. Isaacson, C. Krause, F. Maltoni, O. Mattelaer, and T. Plehn, (2022), arXiv:2212.06172 [hep-ph].
[23] D. Maître and R. Santos-Mateos, (2022), arXiv:2211.02834 [hep-ph].
[24] R. Verheyen, SciPost Phys. **13**, 047 (2022), arXiv:2205.01697 [hep-ph].
[25] A. Butter, T. Plehn, S. Schumann, *et al.*, in *2022 Snowmass Summer Study* (2022) arXiv:2203.07460 [hep-ph].
[26] S. Amoroso *et al.* (HSF Physics Event Generator WG), Comput. Softw. Big Sci. **5**, 12 (2021), arXiv:2004.13687 [hep-ph].
[27] S. Amoroso *et al.* (HSF Physics Event Generator WG), (2020), arXiv:2004.13687 [hep-ph].
[28] HSF Physics Event Generator WG (HSF Physics Event Generator WG), (2021), arXiv:2109.14938 [hep-ph].
[29] J. M. Campbell *et al.*, in *2022 Snowmass Summer Study* (2022) arXiv:2203.11110 [hep-ph].

[30] J. M. Campbell, R. Ellis, and D. L. Rainwater, Phys. Rev. **D68**, 094021 (2003), hep-ph/0308195.
[31] J. M. Campbell and R. K. Ellis, Phys. Rev. **D60**, 113006 (1999), hep-ph/9905386.
[32] J. M. Campbell, R. K. Ellis, and C. Williams, JHEP **07**, 018 (2011), arXiv:1105.0020 [hep-ph].
[33] J. Campbell and T. Neumann, JHEP **12**, 034 (2019), arXiv:1909.09117 [hep-ph].
[34] W. Jakob, "nanobind – seamless operability between c++17 and python," (2022), https://github.com/wjakob/nanobind.
[35] M. Abadi *et al.*, "TensorFlow, Large-scale machine learning on heterogeneous systems," (2015).
[36] E. Bothmann *et al.* (Sherpa), SciPost Phys. **7**, 034 (2019), arXiv:1905.09127 [hep-ph].
[37] S. Catani and M. H. Seymour, Nucl. Phys. **B485**, 291 (1997), hep-ph/9605323.
[38] S. Catani, S. Dittmaier, M. H. Seymour, and Z. Trocsanyi, Nucl. Phys. **B627**, 189 (2002), hep-ph/0201036.
[39] F. Krauss, R. Kuhn, and G. Soff, JHEP **02**, 044 (2002), hep-ph/0109036.
[40] S. Frixione, Z. Kunszt, and A. Signer, Nucl. Phys. **B467**, 399 (1996), hep-ph/9512328.
[41] R. Ellis, K. Melnikov, and G. Zanderighi, JHEP **04**, 077 (2009), arXiv:0901.4101 [hep-ph].
[42] M. Grazzini, S. Kallweit, and M. Wiesemann, Eur. Phys. J. C **78**, 537 (2018), arXiv:1711.06631 [hep-ph].
[43] T. Müller, B. McWilliams, F. Rousselle, M. Gross, and J. Novák, (2018), arXiv:1808.03856 [cs.LG].
[44] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, (2019), arXiv:1906.04032 [stat.ML].
[45] L. Dinh, D. Krueger, and Y. Bengio, (2014), arXiv:1410.8516 [cs.LG].
[46] L. Dinh, J. Sohl-Dickstein, and S. Bengio, (2016), arXiv:1605.08803 [cs.LG].
[47] T. Gleisberg, S. Höche, F. Krauss, A. Schälicke, S. Schumann, and J. Winter, JHEP **02**, 056 (2004), hep-ph/0311263.
[48] T. Gleisberg, S. Höche, F. Krauss, M. Schönherr, S. Schumann, F. Siegert, and J. Winter, JHEP **02**, 007 (2009), arXiv:0811.4622 [hep-ph].
[49] R. D. Ball *et al.* (NNPDF), JHEP **04**, 040 (2015), arXiv:1410.8849 [hep-ph].
[50] M. Cacciari, G. P. Salam, and G. Soyez, JHEP **04**, 063 (2008), arXiv:0802.1189 [hep-ph].
[51] M. Cacciari, G. P. Salam, and G. Soyez, Eur.Phys.J. **C72**, 1896 (2012), arXiv:1111.6097 [hep-ph].
[52] S. Frixione, Phys. Lett. **B429**, 369 (1998), hep-ph/9801442.
[53] J. M. Campbell, S. Höche, and C. T. Preuss, Eur. Phys. J. C **81**, 1117 (2021), arXiv:2107.04472 [hep-ph].
[54] E. Bothmann, W. Giele, S. Höche, J. Isaacson, and M. Knobbe, (2021), 10.21468/SciPostPhysCodeb.3, arXiv:2106.06507 [hep-ph].
[55] A. Valassi, S. Roiser, O. Mattelaer, and S. Hageböck, EPJ Web Conf. **251**, 03045 (2021), arXiv:2106.12631 [physics.comp-ph].
[56] A. Valassi, T. Childers, L. Field, S. Hageböck, W. Hopkins, O. Mattelaer, N. Nichols, S. Roiser, and D. Smith, PoS **ICHEP2022**, 212 (2022), arXiv:2210.11122 [physics.comp-ph].
[57] E. Bothmann, J. Isaacson, M. Knobbe, S. Höche, and W. Giele, PoS **ICHEP2022**, 222 (2022).
[58] K. Danziger, T. Janßen, S. Schumann, and F. Siegert, SciPost Phys. **12**, 164 (2022), arXiv:2109.11964 [hep-ph].
[59] T. Janßen, D. Maître, S. Schumann, F. Siegert, and H. Truong, (2023), arXiv:2301.13562 [hep-ph].
[60] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, Advances in neural information processing systems **32** (2019).
[61] H. Carter Edwards , Christian R. Trott , Daniel Sunderland, JPDC **74**, 3202 (2014).