

Monte Carlo Event Generation with Continuous Normalizing Flows

E. Bothmann^{1,2}, T. Janßen^{2,3,*}, M. Knobbe⁴, B. Schmitzer^{3,5} and F. Sinz^{3,5}

¹*IT Department, CERN, 1211 Geneva 23, Switzerland*

²*Institute for Theoretical Physics, University of Göttingen, Göttingen, Germany*

³*Campus Institute Data Science, University of Göttingen, Göttingen, Germany*

⁴*Fermi National Accelerator Laboratory, Batavia, USA*

⁵*Institute for Computer Science, University of Göttingen, Göttingen, Germany*

 (Received 25 July 2025; revised 11 January 2026; accepted 27 March 2026; published 27 August 2026)

We apply continuous normalizing flows trained with the flow matching method to the problem of phase-space sampling in Monte Carlo event generation for high-energy collider physics. Focusing on lepton-pair and top-quark pair production with multiple jets, the two computationally most expensive processes at the Large Hadron Collider, we train helicity-conditioned continuous normalizing flows to remap the random numbers used in matrix element evaluation. Compared to standard methods, we achieve unweighting efficiency improvements by factors of up to 184 and 25 for the two processes at their respective highest jet number, at the cost of an increased evaluation time. When combining the advantages of continuous normalizing flows with the fast evaluation times of coupling-layer-based flows, using the REGFLOW approach, we find parton-level unweighted event generation walltime gains of about a factor of 10 at the highest jet numbers. These substantial gains highlight the promise of samplers based on machine learning for next-generation collider experiments.

DOI: [10.1103/jt6y-h375](https://doi.org/10.1103/jt6y-h375)

Introduction—The exceptional experimental precision achieved by the ATLAS [1] and CMS [2] experiments during the current and upcoming runs at the Large Hadron Collider (LHC), and its potential successors [3–5] demand equally precise theoretical simulations. Without any improvements over current techniques, the uncertainties in many measurements will be dominated not by experimental statistics, but by deficiencies in Monte Carlo (MC) event samples that underpin critical analyses [6–10]. An important example is vector boson plus jets production, for which very large samples are an essential input for many precision measurements, e.g., for Higgs boson [11,12] and top quark measurements [13,14]. These simulations require improvements in three main areas: parametric accuracy, numerical stability, and statistical precision. This Letter addresses the third, i.e., how to efficiently generate large-scale MC event samples with a statistical quality that matches or exceeds that of experimental data samples. The ATLAS Collaboration estimates that approximately 330 billion simulated events will be needed in the next phase of the LHC to accurately model vector-boson production in association

with additional jets [15], a dominant background process in high-energy analyses. Generating this dataset using current methods would demand about 1000 Perlmutter 2 $\times$ CPU nodes for an entire year [15].

The main bottleneck is evaluating matrix elements for hard-scattering processes, especially at high final-state multiplicities. Although only computationally feasible at leading order, these processes dominate the cost due to their complexity and low unweighting efficiency ϵ [16]. The term unweighting refers to using rejection sampling in order to replace a weighted sample with a typically much smaller unit-weight sample that follows the same underlying probability distribution. This is used in high-energy physics to reduce the cost of storage and downstream simulation steps. The unweighting efficiency ϵ is given by the number of events in the unweighted sample divided by the number of events in the original weighted sample. The main event generators used by LHC experiments are based on multi-channel methods [17] and adaptive importance sampling algorithms like VEGAS [18–20]. For seven final-state particles, one typically finds $\epsilon < 0.01\%$ [21].

Modern machine learning techniques offer an alternative. In particular, normalizing flows (NFs) [22–24] have been studied as drop-in replacements for VEGAS, offering more flexible function approximations [25–32]. Improvements of up to a factor of 10 in efficiency were demonstrated for low to moderate final-state multiplicities. However, for the highest-multiplicity processes that dominate computational budgets, substantial gains have remained elusive.

*Contact author: tjansse2@uni-goettingen.de

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI. Funded by SCOAP³.

In this Letter, we propose—for the first time—the use of continuous normalizing flows (CNFs) [33] trained with the flow matching method [34–37] to solve this problem. We evaluate our approach on the two most computationally intensive processes simulated for the LHC: lepton-pair production and top-quark pair production with multiple jets. We compare performance using the unweighting efficiency ϵ as our key metric, benchmarking against VEGAS-based methods and normalizing flows based on coupling layers [24,38,39]. Our results demonstrate a significant increase in ϵ : for the highest-multiplicity processes, CNFs improve ϵ by factors of up to 184, compared to the other methods. This is enabled in part by conditioning on helicity configurations, which allows the model to learn correlations between discrete and continuous features. Additional details are presented in Ref. [40].

Phase-space sampling—In collider simulations, the primary quantity of interest is the scattering cross section, which measures the probability of a given scattering process to occur. For hadronic collisions, e.g., at the LHC, it is given by

$$\sigma_{h_1 h_2 \rightarrow X} = \sum_{i,j} \int_0^1 dx_1 \int_0^1 dx_2 f_i(x_1, \mu_F) f_j(x_2, \mu_F) \times \hat{\sigma}_{ij \rightarrow X}(x_1, x_2, \mu_R, \mu_F), \quad (1)$$

where the sum runs over partons in the incoming hadrons $h_{1,2}$, i.e. over quarks, antiquarks, and gluons. The parton density functions (PDFs) $f_{i,j}$ are usually provided as interpolation grids in x and μ_F^2 [41]. They vary nonlinearly over many orders of magnitude. The partonic cross section $\hat{\sigma}_{ij \rightarrow X}$ involves an integral over the final-state four-momenta $p_f = (E_f, \vec{p}_f)$ with $f = 3, \dots, m$ and m being the total number of incoming and outgoing particles:

$$d\hat{\sigma}_{ij \rightarrow X} = \frac{1}{2E_1 E_2 |\vec{v}_1 - \vec{v}_2|} \left(\prod_f \frac{d^3 \vec{p}_f}{(2\pi)^3} \frac{1}{2E_f} \right) \times |\mathcal{M}_{ij \rightarrow X}(p_1, p_2 \rightarrow \{p_f\})|^2 \times (2\pi)^4 \delta^{(4)} \left(p_1 + p_2 - \sum_{k=3}^m p_k \right) \Theta(\{p_f\}). \quad (2)$$

The cut function Θ , implemented in terms of Heaviside functions, enforces phase-space cuts imposed by experiment or theory. The squared matrix element $|\mathcal{M}_{ij \rightarrow X}|^2$ can be expressed via helicity amplitudes. The Lorentz-invariant phase space includes a delta function enforcing four-momentum conservation, which reduces the dimensionality of the integral from $3n_{\text{out}}$ to $3n_{\text{out}} - 4$, with $n_{\text{out}} = m - 2$ denoting the number of final-state particles. Including $x_{1,2}$, the total dimensionality becomes $d = 3n_{\text{out}} - 2$, which can reach around 20 for typical LHC simulations.

The integral σ in Eq. (1) defines the overall normalization and can be approximated to the required precision by

MC integration. While the relative error of this integral estimate is also of interest, the real challenge is to generate phase-space samples $x_{1,2} \cup \{\vec{p}_f\}$ distributed as $d\sigma$. These samples must contain enough events for robust comparisons across a large number of observables of interest, ranging across many orders of magnitude in $d\sigma$. The difficulty of the task is increased because the matrix elements $|\mathcal{M}|^2$ are usually sharply peaked and multimodal, and because of the discontinuities introduced by the phase-space cuts Θ . Moreover, the integration variables are often strongly correlated.

To simplify the notation, let p be a probability density function defined by $p(x)dx = \sigma^{-1}d\sigma$. To make the problem more suitable for MC sampling, the function p is pulled back from the manifold of physically valid phase-space points, M , to the unit hypercube $U = [0, 1]^d$ using a bijective map $\phi: U \rightarrow M$, such that

$$p_0(x) = \phi^* p(x) = p(\phi(x)) \det \left[\frac{\partial \phi}{\partial x}(x) \right]. \quad (3)$$

The map ϕ implements four-momentum conservation and Lorentz invariance. Ideally, it simplifies the problem by using physical knowledge of the scattering process under consideration, in particular by flattening the distribution. Rejection sampling is used to sample from p . First, a trial event is generated by sampling from the uniform distribution u_d on U and applying the map ϕ . The density of such events is

$$q(x) = \phi_* u_d(x) = \det \left[\frac{\partial \phi^{-1}}{\partial x}(x) \right]. \quad (4)$$

The trial event is accepted with probability

$$p_{\text{accept}}(x) = \frac{w(x)}{w_{\text{max}}}, \quad \text{where } w(x) := \frac{p(x)}{q(x)}, \quad (5)$$

and rejected otherwise. Using $w_{\text{max}} := \max_x w(x)$, the accepted events follow p . The resulting unweighting efficiency ϵ is given by the average acceptance probability,

$$\epsilon := \frac{\langle w \rangle}{w_{\text{max}}}, \quad (6)$$

which is a crucial figure of merit to measure the performance of a phase-space generator.

In practice, the maximal weight w_{max} needs to be estimated in an initial survey phase from a finite set of events. To mitigate the impact of large outliers, one usually defines an effective $w_{\text{max,eff}}$ to increase the efficiency and to render its value more numerically stable. Events with a weight $w > w_{\text{max,eff}}$ are assigned a non-uniform overweight $w/w_{\text{max,eff}}$. Commonly, $w_{\text{max,eff}}$ is defined to fix the fraction of overweight events via their contribution to the integral,

where e.g. $\epsilon_{0.001}$ denotes the unweighting efficiency for which maximally 0.1% of the integral is contributed by events with overweights.

To increase the unweighting efficiency, automatic optimization techniques can be an effective alternative to manually tuning ϕ . To implement these, ϕ is combined with a second bijective map, $\psi_\theta: U \rightarrow U$, a parametric model with parameters θ that can be adapted to data. Let q_θ be the pushforward of the uniform distribution u_d by ψ_θ ,

$$q_\theta = (\psi_\theta)_* u_d. \quad (7)$$

Sampling from q_θ , the unweighting efficiency becomes

$$\epsilon_\theta = \frac{\langle w_\theta \rangle}{w_{\max, \theta}} \quad \text{with} \quad w_\theta(x) := \frac{p_0(x)}{q_\theta(x)}. \quad (8)$$

In order to maximize the unweighting efficiency, ψ_θ needs to be optimized such that q_θ becomes close to p_0 .

A simple example for ψ_θ is VEGAS, which is a fully factorized approach. It constructs ψ_θ as a product deformation map based on one-dimensional piecewise linear maps. When p_0 is not factorizable, this approach is clearly limited. A more expressive alternative is given by NFs, which use neural networks to parametrize ψ_θ . Through restrictions in their architecture, NFs can be designed in a way so that their Jacobian determinant can easily be evaluated without having to invert the neural networks. Flows based on coupling layers, for example, are implemented as a chain of discrete, simpler steps. In this Letter, we propose to realize ψ_θ as a CNF, where the map is constructed implicitly by integrating a time-dependent vector field.

Continuous normalizing flows—As discussed above, we aim to sample from a target density $p_0: \mathbb{R}^d \rightarrow \mathbb{R}$, but only have access to a latent density q_0 . We seek a map $\psi: \mathbb{R}^d \rightarrow \mathbb{R}^d$ that transforms $x \sim q_0$ into $x' \sim p_0$, or approximately so. NFs provide such a map as a trainable diffeomorphism. Different NF methods vary in construction and training objective. One example are discrete-time flows defined as compositions $\psi = \psi_k \circ \dots \circ \psi_1$, often built using Coupling Layers and trained via KL loss [42]. We instead realize ψ as a CNF—a continuous-time analog—which, as shown below, trains more easily and scales better with dimensionality in our application.

A CNF is defined by a smooth, time-dependent vector field $v_t: [0, 1] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$, which determines a flow $\psi_t: \mathbb{R}^d \rightarrow \mathbb{R}^d$ via the ODE

$$\frac{d\psi_t(x)}{dt} = v_t[\psi_t(x)], \quad (9)$$

with initial condition $\psi_0(x) = x$. Sampling is done by drawing $x_0 \sim q_0$ and integrating

$$x_t = \psi_t(x_0) = \int_0^t v_{t'}(\psi_{t'}(x_0)) dt'. \quad (10)$$

If v_t is Lipschitz in space and continuous in time, the Picard–Lindelöf theorem ensures that ψ_t is bijective.

To evaluate the density $q_t(x_t)$, the following continuity equation is used:

$$\frac{\partial}{\partial t} q_t(x) + \nabla \cdot (q_t(x) v_t(x)) = 0, \quad (11)$$

which leads to

$$\frac{d}{dt} \log q_t(\psi_t(x_0)) + \nabla \cdot v_t(\psi_t(x_0)) = 0. \quad (12)$$

Thus, sampling and density evaluation reduce to solving the joint system

$$\frac{d}{dt} \begin{bmatrix} \psi_t(x_0) \\ \log q_t(\psi_t(x_0)) \end{bmatrix} = \begin{bmatrix} v_t(\psi_t(x_0)) \\ -\nabla \cdot v_t(\psi_t(x_0)) \end{bmatrix} \quad (13)$$

with initial conditions

$$\begin{bmatrix} \psi_t(x_0) \\ \log q_t(x_0) \end{bmatrix}_{t=0} = \begin{bmatrix} x_0 \\ \log q_0(x_0) \end{bmatrix}. \quad (14)$$

Note that we define ψ_t on $\mathbb{R}^d$ rather than the unit cube U to avoid boundary constraints on v_t , naturally choosing a standard normal base distribution. To map outputs to U , an element-wise sigmoid transform is applied, adjusting densities via the Jacobian determinant.

Flow matching—We want to learn v_t (or ψ_t) so that the model density q_1 matches the target p . To this end, we parametrize the vector field as $v_{t,\theta}$ via a neural network with trainable parameters θ . Assuming access to samples $x_1 \sim p(x)$, we aim to adapt $v_{t,\theta}$. One option is maximum likelihood estimation via KL minimization as in [33]. However, this is slow because each training step requires integrating the reverse ODE to evaluate the density.

This motivates simulation-free alternatives [34–37], which directly match $v_{t,\theta}$ to a target vector field u_t generating $p(x)$. There are various ways to construct an admissible u_t . We briefly outline the strategy of Refs. [34,43] via conditional vector fields. Let

$$u_t(x|x_0, x_1) = x_1 - x_0, \quad (15)$$

with $x_0 \sim q_0$ and $x_1 \sim p$ being samples from the base and target distribution, respectively. Clearly, a particle starting at x_0 will flow to x_1 along a straight line when following $u_t(\cdot|x_0, x_1)$, i.e.,

$$x_t = tx_1 + (1-t)x_0. \quad (16)$$

Let now

$$u_t(x) = \mathbb{E}_{(x_0, x_1) \sim \pi, x_t = x} [u_t(x|x_0, x_1)]. \quad (17)$$

It can be shown [34,37,43] that the flow according to u_t transforms q_0 into p as long as the marginals of the joint law π are q_0 and p . For instance, we can set $\pi(x_0, x_1) = q_0(x_0) \otimes p(x_1)$ (independent coupling). Note that we condition the expectation to pairs (x_0, x_1) where x_t , Eq. (16), moves through the query point x . It is now natural to fit $v_{t,\theta}(x)$ to $u_t(x)$, e.g., via

$$\mathbb{E}_{(x_0, x_1) \sim \pi, t \sim U_1} \|v_{t,\theta}(x_t) - u_t(x_t)\|^2. \quad (18)$$

Note that evaluating u_t at specified points x is cumbersome due to the conditioning in Eq. (17). Fortunately, one can show [34,43] that Eq. (18) has the same minimizers as the flow matching objective

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{(x_0, x_1) \sim \pi, t \sim U_1} \|v_{t,\theta}(x_t) - u_t(x_t|x_0, x_1)\|^2. \quad (19)$$

It is local in space and time, so it can be evaluated fast and without solving an ODE. Also, in contrast to a KL loss, the minimizer is unique [in terms of $v_{t,\theta}(x)$], which should lead to more efficient use of the model parameters. More generally, one can add noise to the interpolation to increase robustness [34,43]. This is done by adding noise with a small standard deviation σ_{noise} to the individual straight lines x_t . This way, the sampled data cover more volume. We use $\sigma_{\text{noise}} = 10^{-4}$.

We make two modifications to the objective. First, since the data lies in the unit hypercube U , the inverse of the sigmoid transform—namely, the logit function—is used to map it to $\mathbb{R}^d$. To avoid numerical issues, we combine it with the affine transform $x \mapsto (1 - \epsilon)x + \epsilon/2$ with $\epsilon = 10^{-6}$. Second, we cannot sample $x_1 \sim p$ directly; instead, we sample from our model distribution $q_1 = (\psi_\theta)_* q_0$. To account for the mismatch in density, the loss [Eq. (19)], is multiplied by the importance weight w , resulting in a variant of energy conditional flow matching [43]. Since high variance in the weights can impair training in high dimensions, choosing a map ϕ that minimizes variance is crucial. After each training iteration, the model can generate samples to allow further iterative refinement. If available, a pretrained VEGAS can also provide initial samples, enhancing the overall training process.

Application—We apply our neural network optimization method to dominant partonic channels in standard-candle processes at the LHC: lepton and top pair production with additional jets at $\sqrt{s} = 14$ TeV. We focus on the channels $d\bar{d} \rightarrow e^+e^- + ng$ and $gg \rightarrow t\bar{t} + ng$, which contribute the largest cross sections. We use the CHILI phase space to define the mapping ϕ which maps the unit hypercube to the physical phase space [44]. The CHILI mapping is simple yet effective, using a single integration channel that performs competitively with complex multichannel-based methods

for several standard LHC processes, including the ones studied in this Letter [45].

To evaluate the matrix elements, we use PEPPER [45–47], a parton-level generator optimized for complex LHC processes with GPU acceleration. It employs an internal implementation of CHILI and is integrated in the LHC simulation toolchain by interfaces to the widely used event generators SHERPA [48–50] and PYTHIA [51–54]. Matrix elements are evaluated in PEPPER by summing over color indices using a minimal color decomposition [55–57], while nonvanishing helicity configurations are sampled. Normally, the helicity sampling is adapted to minimize variance, similar to adding a discrete optimization dimension to VEGAS. Under neural network optimization, we jointly optimize phase-space and helicity sampling, exploiting correlations between them, by introducing the helicities as a conditioning variable for the networks. For this Letter we have added a simple and straightforward file-based interface to PEPPER for training and evaluating machine learning models, based on the scalable LHEF5 format [58,59].

For the PDFs, we use the NNPDF3.0 [60] set via LHAPDF6 [41]. The renormalization and factorization scales are set to $\mu_R^2 = \mu_F^2 = H_T^2/2$ for lepton pair production, and to $H_T^2/2$ for top-quark pair production [61]. The following electroweak parameters are used: $\sin^2\theta_w = 0.23155$, $\alpha = 1/128.80$, $m_Z = 91.1876$ GeV, $\Gamma_Z = 2.4952$ GeV, and the top-quark mass $m_t = 173.21$ GeV. All other quarks are massless. For lepton pairs we additionally require $66 \text{ GeV} \leq m_{e^+e^-} \leq 116 \text{ GeV}$. For all massless partons we enforce the additional jet cuts $p_{T,j} > 30 \text{ GeV}$, $|\eta_j| < 5$, and $\Delta R_{ij} > 0.4$.

All VEGAS grids have 100 bins per dimension, optimized over 15 steps. For the highest multiplicity, we use $\sim 3 \times 10^8$ points, yielding 2.5×10^5 training points per nonvanishing helicity configuration in lepton-pair production and 8×10^4 for top-quark pair production. The NFs are optimized in 8 iterations. Initially, events are generated without remapping. In subsequent steps, trained flows produce inputs used by PEPPER to generate training data. An embedding layer encodes helicity configurations. Our coupling flows use the minimal number of layers needed to capture correlations, with transformations based on multilayer perceptrons (MLPs), and an early stopping criterion based on validation loss. Our ODE Flows use MLP-based vector fields with time encoded via Fourier features [62]. Both models are trained using the ADAMW optimizer [63]. After training, the models are frozen and used to generate weighted events.

Results—We report relative improvements in the unweighting efficiency $\epsilon_{0.001}$, which serves as the primary performance metric of this Letter. Figure 1 illustrates the results for the processes $d\bar{d} \rightarrow e^+e^- + ng$ and $gg \rightarrow t\bar{t} + ng$, comparing the coupling flow and the ODE flow mappings to VEGAS. We observe that ODE flows exhibit a favorable scaling of $\epsilon_{0.001}$ with increasing n ,

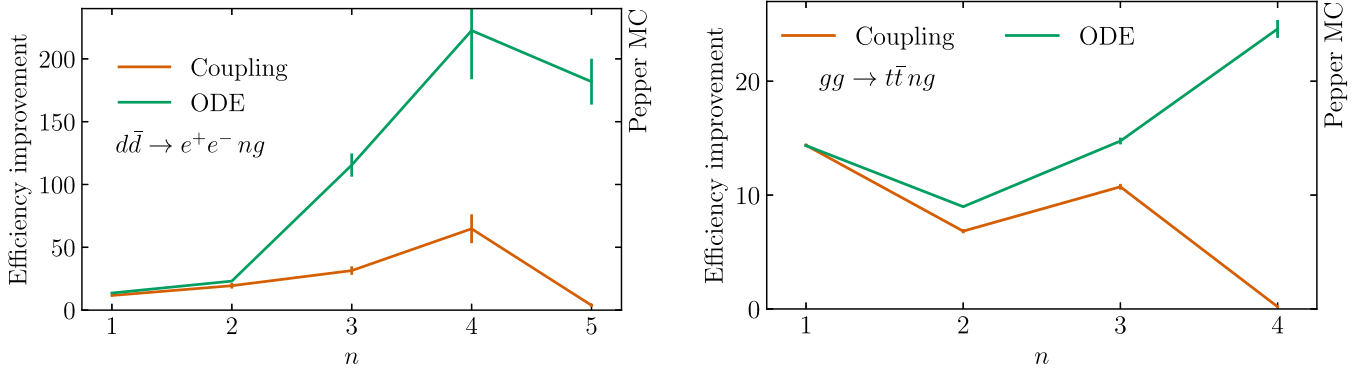


FIG. 1. Relative improvements for the unweighting efficiencies $\epsilon_{0.001}$, both for $e^+e^- + n$ gluons (left) and for $t\bar{t} + n$ gluons (right), as a function of n . The improvements are shown for the coupling flows (“coupling”), and ODE flows (“ODE”), compared to VEGAS. Each curve represents the mean over ten independent evaluations of the unweighting efficiency with corresponding statistical errors.

outperforming the other methods as the complexity grows. In contrast, the performance of coupling flows deteriorates at the highest multiplicity studied, with efficiency falling below that of the VEGAS baseline in the $gg \rightarrow t\bar{t} + 4g$ case. For $d\bar{d} \rightarrow e^+e^- + 5g$, the ODE flow achieves $\epsilon_{0.001} = 1.29(8)\%$, which is about $43\times$ higher than the result for the coupling flow and $184\times$ higher than the one for VEGAS. For $\epsilon_{0.01}$, the relative gains are 13 and 153, respectively. For $gg \rightarrow t\bar{t} + 4g$, the ODE Flow yields $\epsilon_{0.001} = 5.76(9)\%$, outperforming the coupling flow by a factor of 144 and VEGAS by a factor of 25. For $\epsilon_{0.01}$, the respective relative gains are 8 and 17.

Unweighting efficiency improvements of the ODE flow over VEGAS are consistently greater for $d\bar{d} \rightarrow e^+e^- + n g$ than for $gg \rightarrow t\bar{t} + n g$. At $n = 4$, we see a factor 198 improvement of $\epsilon_{0.001}$ for the former, versus a factor 25 for the latter. This is due to a better VEGAS baseline performance for top-pair production, visible also in the relative integration errors we find. This suggests structural differences in phase-space factorization—e.g., due to the Z-boson resonance and the stronger dependence on the helicity configuration in the lepton-pair production case.

Our findings demonstrate that flow matching outperforms alternative approaches, in particular in terms of the unweighting efficiency—the key metric for efficient event generation. Its advantage over coupling flows is largest at high multiplicities, where computational cost is highest, making it a highly promising candidate for current and future LHC simulation campaigns.

Although the ODE-based models incur longer inference times, their performance can be efficiently transferred to fast coupling flows using the recently proposed REGFlow method [64], which trains the discrete model on pairs generated by the ODE Flow. In our benchmarks, REGFlow-trained coupling flows recover a sizable fraction of the ODE efficiencies while maintaining 2 orders of magnitude faster inference, yielding effective walltime gains of a factor of 12 for $d\bar{d} \rightarrow e^+e^- + 5g$ and 8 for $gg \rightarrow t\bar{t} + 4g$ when compared to VEGAS. The longer training is

offset by the speed-up after two million events for $d\bar{d} \rightarrow e^+e^- + 5g$, and after 40 million events for $gg \rightarrow t\bar{t} + 4g$.

Conclusion—We presented the first application of flow matching to the problem of high-dimensional phase-space sampling in high-energy physics. Our approach jointly remaps the input random numbers used to select continuous kinematic variables and discrete helicity configurations, enabling more accurate and efficient sampling of complex final states. A simple, file-based interface implemented in PEPPER, built around the LHEH5 format, both facilitates the external optimization, and the utilization of our results for further downstream simulation steps using established event generation frameworks such as SHERPA and PYTHIA. Focusing on two challenging benchmark processes, lepton-pair and top-antitop pair production with up to five and four associated jets, respectively, we demonstrated unweighting efficiency improvements of up to $184\times$ and $25\times$ over VEGAS baseline results. We further showed that a significant fraction of the efficiency gains achieved with flow-matching-based ODE flows can be transferred effectively to fast coupling-layer-based flows using the REGFlow method, yielding event generation walltime reductions of around a factor of 10 at the highest jet numbers studied. This shows that the efficiency of coupling-layer-based flows can be substantially improved by effectively using a flow matching objective instead of the maximum-likelihood objective otherwise used to train them in this and previous studies. We further expect that ODE flows themselves will become significantly faster as these architectures mature and more efficient implementations are developed.

While our Letter concentrated on one dominant partonic channel for each process, full simulations will require all contributing channels. Future work will extend the method to a single, conditional model that simultaneously learns across multiple partonic channels and jet multiplicities, leveraging interchannel correlations for further gains in sampling efficiency. The sampling method improved in this way will be made available as part of a future public PEPPER

release, contributing to a successful high-luminosity LHC and future collider physics programs.

Acknowledgments—The authors gratefully acknowledge the computing time granted by the Resource Allocation Board and provided on the supercomputer Emmy/Grete at NHR-Nord@Göttingen as part of the NHR infrastructure. The calculations for this research were conducted with computing resources under the project nhr_ni_starter_22045. The authors also acknowledge the use of computing resources made available by CERN to conduct some of the research reported in this work. This material is based upon work supported by Fermi Forward Discovery Group, LLC under Contract No. 89243024CSC000002 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics. E. B. and M. K. acknowledge support by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) No. 510810461. T. J. expresses his gratitude to CIDAS for supporting him with a fellowship and acknowledges financial support from the German Federal Ministry of Education and Research (BMBF; now Federal Ministry of Research, Technology and Space, BMFTR) in the ErUM-Data action plan through the KISS consortium (Verbundprojekt No. 05D2022). B. S. and F. S. were supported by the German Research Foundation (DFG) SFB 1456, Mathematics of Experiment—Project-ID No. 432680300. B. S. was supported by the Emmy Noether Programme of the DFG (Project-ID No. 403056140).

Data availability—The data that support the findings of this article are openly available [65].

- [1] G. Aad *et al.* (ATLAS Collaboration), The ATLAS experiment at the CERN large hadron collider, *J. Instrum.* **3**, S08003 (2008).
- [2] S. Chatrchyan *et al.* (CMS Collaboration), The CMS experiment at the CERN LHC, *J. Instrum.* **3**, S08004 (2008).
- [3] A. Abada *et al.* (FCC Collaboration), FCC physics opportunities: Future circular collider conceptual design report volume 1, *Eur. Phys. J. C* **79**, 474 (2019).
- [4] A. Abada *et al.* (FCC Collaboration), FCC-ee: The lepton collider: Future circular collider conceptual design report volume 2, *Eur. Phys. J. Spec. Top.* **228**, 261 (2019).
- [5] A. Abada *et al.* (FCC Collaboration), FCC-hh: The hadron collider: Future circular collider conceptual design report volume 3, *Eur. Phys. J. Spec. Top.* **228**, 755 (2019).
- [6] A. Buckley *et al.*, General-purpose event generators for LHC physics, *Phys. Rep.* **504**, 145 (2011).
- [7] European Strategy Group, *2020 Update of the European Strategy for Particle Physics* (CERN Council, Geneva, 2020), [10.17181/ESU2020](https://arxiv.org/abs/10.17181/ESU2020).
- [8] M. Narain *et al.*, The future of US particle physics—The snowmass 2021 energy frontier report, [arXiv:2211.11084](https://arxiv.org/abs/2211.11084).
- [9] J. Albrecht *et al.* (HEP Software Foundation Collaboration), A roadmap for HEP software and computing R&D for the 2020s, *Comput. Software Big Sci.* **3**, 7 (2019).
- [10] S. Amoroso *et al.* (HSF Physics Event Generator WG Collaboration), Challenges in Monte Carlo event generator software for high-luminosity LHC, *Comput. Software Big Sci.* **5**, 12 (2021).
- [11] G. Aad *et al.* (ATLAS Collaboration), Measurement of the production cross section for a Higgs boson in association with a vector boson in the $H \rightarrow WW^* \rightarrow \ell\nu\ell\nu$ channel in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector, *Phys. Lett. B* **798**, 134949 (2019).
- [12] G. Aad *et al.* (ATLAS Collaboration), Measurement of the associated production of a Higgs boson decaying into b -quarks with a vector boson at high transverse momentum in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector, *Phys. Lett. B* **816**, 136204 (2021).
- [13] M. Aaboud *et al.* (ATLAS Collaboration), Measurements of top-quark pair spin correlations in the $e\mu$ channel at $\sqrt{s} = 13$ TeV using pp collisions in the ATLAS detector, *Eur. Phys. J. C* **80**, 754 (2020).
- [14] G. Aad *et al.* (ATLAS Collaboration), Measurement of the $t\bar{t}$ production cross-section in the lepton + jets channel at $\sqrt{s} = 13$ TeV with the ATLAS experiment, *Phys. Lett. B* **810**, 135797 (2020).
- [15] G. Aad *et al.* (ATLAS Collaboration), Modelling and computational improvements to the simulation of single vector-boson plus jet processes for the ATLAS experiment, *J. High Energy Phys.* **8** (2022) 089.
- [16] E. Bothmann, A. Buckley, I. A. Christidi, C. Gütschow, S. Höche, M. Knobbe, T. Martin, and M. Schönherr, Accelerating LHC event generation with simplified pilot runs and fast PDFs, *Eur. Phys. J. C* **82**, 1128 (2022).
- [17] R. Kleiss and R. Pittau, Weight optimization in multi-channel Monte Carlo, *Comput. Phys. Commun.* **83**, 141 (1994).
- [18] G. P. Lepage, A new algorithm for adaptive multidimensional integration, *J. Comput. Phys.* **27**, 192 (1978).
- [19] T. Ohl, VEGAS revisited: Adaptive Monte Carlo integration beyond factorization, *Comput. Phys. Commun.* **120**, 13 (1999).
- [20] G. P. Lepage, Adaptive multidimensional integration: VEGAS enhanced, *J. Comput. Phys.* **439**, 110386 (2021).
- [21] S. Höche, S. Prestel, and H. Schulz, Simulation of vector boson plus many jet final states at the high luminosity LHC, *Phys. Rev. D* **100**, 014024 (2019).
- [22] E. Tabak and E. Vanden-Eijnden, Density estimation by dual ascent of the log-likelihood, *Commun. Math. Sci.* **8**, 217 (2010).
- [23] E. G. Tabak and C. V. Turner, A family of nonparametric density estimation algorithms, *Commun. Pure Appl. Math.* **66**, 145 (2013).
- [24] L. Dinh, D. Krueger, and Y. Bengio, NICE: Non-linear independent components estimation, in *3rd International Conference on Learning Representations, Workshop Track Proceedings* (2015), [arXiv:1410.8516](https://arxiv.org/abs/1410.8516).
- [25] M. D. Klimek and M. Perelstein, Neural network-based approach to phase space integration, *SciPost Phys.* **9**, 053 (2020).

- [26] E. Bothmann, T. Janßen, M. Knobbe, T. Schmale, and S. Schumann, Exploring phase space with neural importance sampling, *SciPost Phys.* **8**, 069 (2020).
- [27] C. Gao, S. Höche, J. Isaacson, C. Krause, and H. Schulz, Event generation with normalizing flows, *Phys. Rev. D* **101**, 076002 (2020).
- [28] T. Heimel, R. Winterhalder, A. Butter, J. Isaacson, C. Krause, F. Maltoni, O. Mattelaer, and T. Plehn, MadNIS—Neural multi-channel importance sampling, *SciPost Phys.* **15**, 141 (2023).
- [29] R. Verheyen, Event generation and density estimation with surjective normalizing flows, *SciPost Phys.* **13**, 047 (2022).
- [30] T. Heimel, N. Huetsch, F. Maltoni, O. Mattelaer, T. Plehn, and R. Winterhalder, The MadNIS reloaded, *SciPost Phys.* **17**, 023 (2024).
- [31] T. Heimel, O. Mattelaer, T. Plehn, and R. Winterhalder, Differentiable MadNIS-lite, *SciPost Phys.* **18**, 017 (2025).
- [32] S. Badger *et al.*, Machine learning and LHC event generation, *SciPost Phys.* **14**, 079 (2023).
- [33] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, Neural ordinary differential equations, in *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (Curran Associates Inc., Red Hook, NY, USA, 2018), pp. 6572–6583, [arXiv:1806.07366](#).
- [34] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, Flow matching for generative modeling, in *The Eleventh International Conference on Learning Representations* (2023), [arXiv:2210.02747](#).
- [35] M. S. Albergo and E. Vanden-Eijnden, Building normalizing flows with stochastic interpolants, in *The Eleventh International Conference on Learning Representations* (2023), [arXiv:2209.15571](#).
- [36] M. S. Albergo, N. M. Boffi, and E. Vanden-Eijnden, Stochastic interpolants: A unifying framework for flows and diffusions, [arXiv:2303.08797](#).
- [37] X. Liu, C. Gong, and Q. Liu, Flow straight and fast: Learning to generate and transfer data with rectified flow, [arXiv:2209.03003](#).
- [38] T. Müller, B. McWilliams, F. Rousselle, M. Gross, and J. Novák, Neural importance sampling, *ACM Trans. Graph.* **38**, 1 (2019).
- [39] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, Neural spline flows, in *Neural spline flows, in Proceedings of the 33rd International Conference on Neural Information Processing Systems (NeurIPS 2019)* (Curran Associates, Inc., Red Hook, NY, 2019).
- [40] E. Bothmann, T. Janßen, M. Knobbe, B. Schmitzer, and F. Sinz, companion paper, Efficient many-jet event generation with flow matching, *Phys. Rev. D* **114**, 034055 (2026).
- [41] A. Buckley, J. Ferrando, S. Lloyd, K. Nordström, B. Page, M. Rüfenacht, M. Schönherr, and G. Watt, LHAPDF6: Parton density access in the LHC precision era, *Eur. Phys. J. C* **75**, 132 (2015).
- [42] S. Kullback and R. A. Leibler, On information and sufficiency, *Ann. Math. Stat.* **22**, 79 (1951).
- [43] A. Tong, K. Fatras, N. Malkin, G. Hugué, Y. Zhang, J. Rector-Brooks, G. Wolf, and Y. Bengio, Improving and generalizing flow-based generative models with minibatch optimal transport, [arXiv:2302.00482](#).
- [44] E. Bothmann, T. Childers, W. Giele, F. Herren, S. Hoeche, J. Isaacson, M. Knobbe, and R. Wang, Efficient phase-space generation for hadron collider event simulation, *SciPost Phys.* **15**, 169 (2023).
- [45] E. Bothmann, T. Childers, W. Giele, S. Höche, J. Isaacson, and M. Knobbe, A portable parton-level event generator for the high-luminosity LHC, *SciPost Phys.* **17**, 081 (2024).
- [46] E. Bothmann, W. Giele, S. Hoeche, J. Isaacson, and M. Knobbe, Many-gluon tree amplitudes on modern GPUs: A case study for novel event generators, *SciPost Phys. Codebases* **2022**, 3 (2022).
- [47] E. Bothmann, J. Isaacson, M. Knobbe, S. Höche, and W. Giele, QCD tree amplitudes on modern GPUs: A case study for novel event generators, *Proc. Sci. ICHEP2022* (**2022**) 222.
- [48] T. Gleisberg, S. Höche, F. Krauss, M. Schönherr, S. Schumann, F. Siegert, and J. Winter, Event generation with SHERPA 1.1, *J. High Energy Phys.* **02** (2009) 007.
- [49] E. Bothmann *et al.* (SHERPA Collaboration), Event generation with SHERPA 2.2, *SciPost Phys.* **7**, 034 (2019).
- [50] E. Bothmann *et al.* (SHERPA Collaboration), Event generation with SHERPA 3, *J. High Energy Phys.* **12** (2024) 156.
- [51] C. Bierlich *et al.*, A comprehensive guide to the physics and usage of PYTHIA 8.3, *SciPost Phys. Codebases* **2022**, 8 (2022).
- [52] T. Sjöstrand, S. Ask, J. R. Christiansen, R. Corke, N. Desai, P. Ilten, S. Mrenna, S. Prestel, C. O. Rasmussen, and P. Z. Skands, An introduction to PYTHIA 8.2, *Comput. Phys. Commun.* **191**, 159 (2015).
- [53] T. Sjostrand, S. Mrenna, and P. Z. Skands, PYTHIA 6.4 physics and manual, *J. High Energy Phys.* **05** (2006) 026.
- [54] T. Sjostrand, S. Mrenna, and P. Z. Skands, A brief introduction to PYTHIA 8.1, *Comput. Phys. Commun.* **178**, 852 (2008).
- [55] T. Melia, Dyck words and multi-quark primitive amplitudes, *Phys. Rev. D* **88**, 014020 (2013).
- [56] T. Melia, Dyck words and multi-quark amplitudes, *Proc. Sci. RADCOR2013* (**2013**) 031.
- [57] H. Johansson and A. Ochirov, Color-kinematics duality for QCD amplitudes, *J. High Energy Phys.* **01** (2016) 170.
- [58] S. Höche, S. Prestel, and H. Schulz, Simulation of vector boson plus many jet final states at the high luminosity LHC, *Phys. Rev. D* **100**, 014024 (2019).
- [59] E. Bothmann, T. Childers, C. Gütschow, S. Höche, P. Hovland, J. Isaacson, M. Knobbe, and R. Latham, Efficient precision simulation of processes with many-jet final states at the LHC, *Phys. Rev. D* **109**, 014013 (2024).
- [60] R. D. Ball *et al.* (NNPDF Collaboration), Parton distributions for the LHC Run II, *J. High Energy Phys.* **04** (2015) 040.
- [61] Z. Bern, L. J. Dixon, F. Febres Cordero, S. Höche, H. Ita, D. A. Kosower, D. Maître, and K. J. Ozeren, Next-to-leading order $W + 5$ -Jet production at the LHC, *Phys. Rev. D* **88**, 014025 (2013).
- [62] M. Tancik, P. P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. T. Barron, and R. Ng, Fourier features let networks learn high frequency functions in low dimensional domains, in *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Curran Associates Inc., Red Hook, NY, USA, 2020), [arXiv:2006.10739](#).

- [63] I. Loshchilov and F. Hutter, Decoupled weight decay regularization, in *Proceedings of the 7th International Conference on Learning Representations* (2019), [arXiv:1711.05101](#).
- [64] D. Rehman, O. Davis, J. Lu, J. Tang, M. Bronstein, Y. Bengio, A. Tong, and A. J. Bose, Efficient regression-based training of normalizing flows for Boltzmann generators, [arXiv:2506.01158](#).
- [65] E. Bothmann, T. Janßen, M. Knobbe, B. Schmitzer, and F. Sinz, Efficient many-jet event generation with flow matching, Zenodo (2026), [10.5281/zenodo.19949391](#).