Normalizing Flows for LHC Theory

Anja Butter
Institut für Theoretische Physik, Universität Heidelberg, Germany
E-mail: butter@thphys.uni-heidelberg.de

Abstract. Over the next years, measurements at the LHC and the HL-LHC will provide us with a wealth of new data. The best hope to answer fundamental questions, like the nature of dark matter, is to adopt big data techniques in simulations and analyses to extract all relevant information. On the theory side, LHC physics crucially relies on our ability to simulate events efficiently from first principles. These simulations will face unprecedented precision requirements to match the experimental accuracy. Innovative ML techniques like generative networks can help us overcome limitations from the high dimensionality of the phase space. Such networks can be employed within established simulation tools or as part of a new framework. Since neural networks can be inverted, they open new avenues in LHC analyses.

1. Introduction

The interpretation of the huge amount of data measured in collider searches depends on the quality of associated precision simulations. A unique strength of high-energy physics is the ability to build first principle based event generators. Starting from a fundamental Lagrangian they are able to generate parton level events followed by parton showers and hadronization. Finally we can simulate the interaction of the final particles with complex detectors. While demonstrating impressive performance, computing resources become a limiting factor for precision simulations. The complexity of loop calculations grows exponentially with the order in coupling strength as well as the number of final state particles. At the same time the mapping of the underlying phase space distributions becomes more challenging due to intermediate resonances and phase space edges, which leads to poor unweighting efficiencies. Finally, full simulations of detector responses are a major bottleneck of the simulations chain. It is therefore crucial to explore new methods to increase the efficiency of the simulation chain, to fully exploit the collider data [1].

While the use of different kinds of neural networks has already been firmly established for parton densities [2], their application for other building blocks of the simulation chain like the estimation of amplitudes [3, 4, 5] is rapidly evolving. Generative networks open new possibilities to enhance the efficiency of simulations [1, 6]. They are able to learn underlying distributions with high precision [7, 8, 9, 10, 11, 12, 13, 14] and can therefore provide more efficient phase space mapping [15, 16, 17, 18, 19, 20], amplify [21, 22] and compress data [23], serve as surrogate models in phenomenological studies and provide fast detector simulations [24, 25, 26]. Finally generative networks enable the inversion of the simulation chain [27, 28, 29]. This contribution discusses the advantages of normalizing flows as a particular suitable type of generative networks for precision simulations and their application to the generation of events in forward and inverse direction.

2. Normalizing flows

Neural network based generative models can be sorted into three main categories: variational autoencoders [30], generative adversarial networks [31] and normalizing flows [32]. While the focus of early studies in event generation and detector simulation was on GANs and VAEs, the inherently high degree of control of normalizing flows has become an essential asset when moving further in the implementation of neural networks for precision simulations and inference.

Normalizing flows consist of multiple bijective layers which induce an invertible mapping f from a latent space Z to a target space X of equal dimension. In order to generate a distribution of events in the target space, the normalizing flow requires as input a distribution in the latent space Z which is typically chosen to be Gaussian. Since the normalizing flow provides a bijective mapping, we can then evaluate the density of the generated distribution $p(x) = p(f(z))$ via

$$p(x) = p(z) \left| \frac{\mathrm{d}x}{\mathrm{d}z} \right|^{-1} , \tag{1}$$

relying on the known density of the multivariate Gaussian in the latent space and the ability to efficiently compute the Jacobian of the flow.

This property of normalizing flows opens up two methods for their training. We can either train a model on explizit density values, given for instance by the amplitude of a process, or we can train on a representative sample of training data, i.e. events, that were drawn from the target distributions. For the training on a target density, we have to minimize the difference between target and generated density, parametrized within the loss function. Suitable choices are given for instance by the mean squared error or the KL divergence via

$$L_{density} = -\left\langle \log \frac{p_{gen}(x)}{p_{true}(x)} \bigg|_{x=f(z)} \right\rangle_{z\sim\mathcal{N}} . \tag{2}$$

As we sample new points during each iteration of the training, this methods relies on our ability to calculate the target density efficiently during training time. Expensive calculations of the target density (for instance through loop-amplitudes) can hence become a bottleneck in precision simulations. An alternative approach learns the target distribution from representative samples. The network is trained in the inverse direction to map these samples to a gaussian distribution in the latent space. The loss is therefore based on the posterior of the network weights θ given training data.

$$\begin{aligned} L_{samples} &= -\log p(\theta|x) && (3)\\ &= -\log p(x|\theta) - \log p(\theta) + \log p(x) && (4)\\ &= -\log p(z|\theta) \cdot \left| \frac{\mathrm{d}x}{\mathrm{d}z} \right|^{-1} - \log p(\theta) + \text{const} . && (5) \end{aligned}$$

In the second line we have applied Bayes' theorem to replace the posterior of the network weights with a conditional probability of the data given network parameters. The remaining terms include a prior on the network parameters and the likelihood of the data which yields only a constant contribution to the loss, as it is independent of the network parameters. In the last line we use the change of variables formula in eq. (1). The resulting conditional probability of the latent variable $z = f^{-1}(x)$ is given by a Gaussian which simplifies to a contribution to the loss proportional to z^2. From the derivation of the loss function it becomes apparent that an efficient training requires not only a bijective network, but also short evaluation time in the inverse direction. In the following we will therefore refer to these networks as invertible neural networks (INN) [33, 34]. Suitable architectures are given for instance by affine coupling layers or cubic splines.

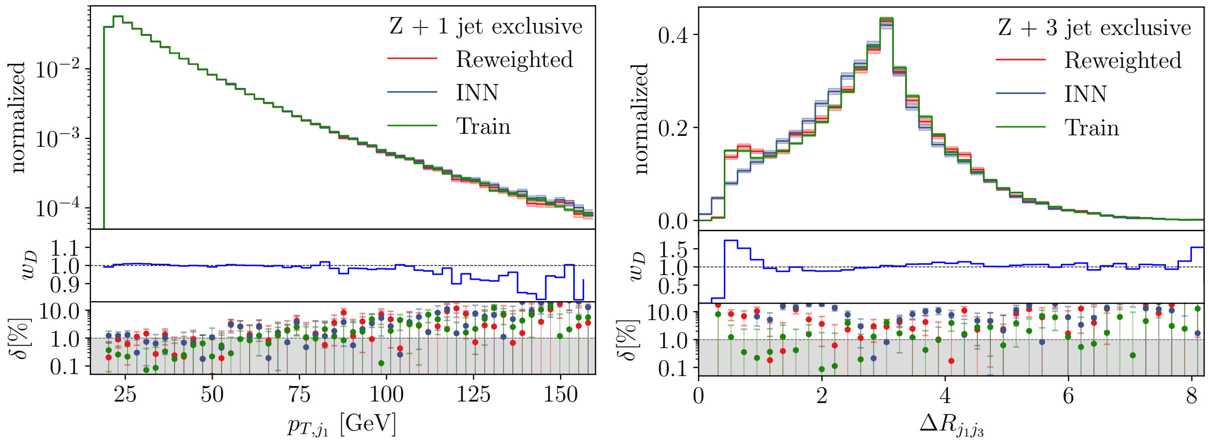

Figure 1. Discriminator-reweighted INN distributions of the jet transverse momenta in the $Z + 1$ jet exclusive channel (left), and the angular distance of jets in the $Z + 3$ jets exclusive channel (right) from a combined $Z+$ jets generation. The bottom panels show the average correction factor obtained from the discriminator output. A detailed description can be found in the original publication [8].

While many advances in machine learning have been achieved through overall improvements of the performance in regression or classification problems, precision simulations make it essential to evaluate in addition the uncertainties of their predictions. By replacing network weights with parametrized weight distributions, Bayesian networks have demonstrated their ability to estimate uncertainties coming from noisy data as well as limited convergence of the network [35, 36]. The corresponding loss is extended to include a prior on the weight distributions, parametrized by their mean and standard deviation. The combination of Bayesian networks with normalizing flows enables uncertainty estimates of generative networks [37]. Sampling from the network weights generates a distribution over distributions which encodes the uncertainty of the network. Furthermore through inversion we can access directly the uncertainty on the density of any phase space point.

3. Precision flows at work

The control over generated phase space densities combined with efficient training due to stable loss functions, have spawned multiple applications of normalizing flows for simulations in high-energy physics.

3.1. Optimizing the phase space sampling

The computation of cross sections for specific processes relies on the numerical integration of amplitudes. While holes and complex phase space structures make naive integration methods highly inefficient, a suitable remapping of the integration variable can lead to nearly constant integrands and increase the efficiency of the integration procedure. The optimal remapping is hence equivalent to the minimization of the density loss in eq. (2) making normalizing flows an ideal choice for the optimization of phase space mapping [15, 16, 17, 18, 19, 20].

3.2. Event generation with uncertainties

Once the calculation of amplitudes becomes expensive, the recycling of training data within a sample based training becomes particularly advantageous. However, without the direct evaluation of the phase space density, we do not have access to exact predictions. It therefore

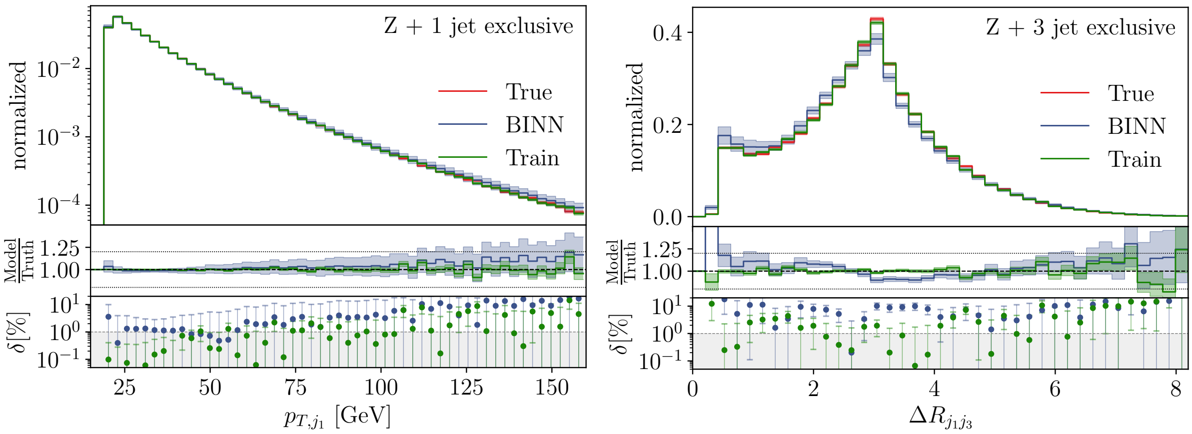

Figure 2. BINN densities and uncertainties of the jet transverse momenta in the $Z+1$ jet exclusive channel (left), and the angular distance of jets in the $Z+3$ jets exclusive channel (right) from a combined $Z+$ jets generation. A detailed description can be found in the original publication [8].

becomes particularly important to achieve high precision with respect to underlying training data and establish a suitable framework to estimate uncertainties.

Fig. 1 shows the results of a standard invertible network trained on Z+jets data at shower level. In the left panel we see that the decreasing distribution of the transverse momenta is captured with high accuracy.

Limitations of the naive INN approach can arise however when the networks is not sufficiently expressive to learn specific features of the dataset. An example is shown in the right panel of Fig. 1, where the INN was unable to learn the topological holes in the distance between two jets induced by the jet algorithm. The treatment of such deviations requires an additional helper network. Classifiers trained to distinguish between generated and training data via a standard binomial cross entropy

$$\mathcal{L} = -\sum_{x\sim p_\text{data}} \log(D(x)) - \sum_{x\sim p_\text{INN}} \log(1-D(x)) \tag{6}$$

$$= -\int \mathrm{d}x\; p_\text{data}(x)\log(D(x)) + p_\text{INN}(x)\log(1-D(x)) \;, \tag{7}$$

learn to estimate the density ratio of generated and training data

$$\frac{p_\text{data}(x)}{p_\text{INN}(x)} = \frac{D(x)}{1-D(x)} \;. \tag{8}$$

The resulting classifier can be used either to estimate systematic deviations between training and generated distributions or directly correct for the difference by assigning a weight $w = \frac{D(x)}{1-D(x)}$ to each generated event x.

The resulting reweighted distributions are indicated by the red lines in Fig. 1. In order to avoid large reweighting factors we can reuse the obtained weighted as feedback in a weighted training [38, 39, 8] to improve the accuracy of the original simulation, resulting in turn in weights closer to 1.

Even when corrections based on the classifier are taken into account, uncertainty estimates via the Bayesian INN remain crucial, to capture effects of limited training statistics, imperfect optimization and fluctuations in the training. Fig. 2 shows how the deviations between generated data and truth are captured by the uncertainties modeled by the Bayesian implementation.

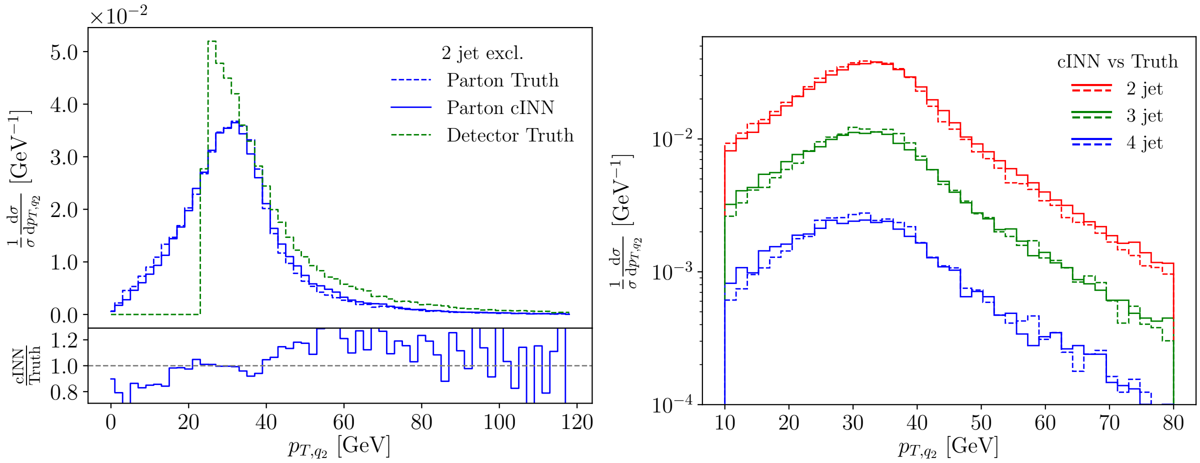

Figure 3. Comparison of truth level and unfolded distributions for 2-jet exclusive (left) and inclusive channels(right). A detailed description can be found in the original publication [29].

3.3. Detector simulations

Full simulations of detector responses model every individual component of a detector, making them highly computing intensive. Normalizing flows have demonstrated promising performance for detector simulations, for instance in the simulation of e^+, γ and π^+ showers [25, 26]. The performance evaluated in terms of a classifier metric demonstrated significant improvements with respect to standard GAN trainings.

4. Inverting the simulation chain with invertible networks

While the conceptual development of precise generative networks including a framework for uncertainty estimates has direct applications in event generation, it also serves as a basis for further applications like the inversion of the simulation chain.

Due to the probabilistic nature of shower and detector simulation, the inversion of the simulation can not be solved with a deterministic mapping. Instead, a given detector level measurement, can be induced by various parton level events configurations. Hence, the inversion needs to be implemented as a conditional event generator that produces a probability distribution over possible parton level configurations for any given detector level measurement [29]. Due to the flexible format of conditional information we can simultaneously train on data with varying number of jets. Fig. 3 shows the unfolded distributions for a Z+jets dataset with up to three hard jets. The left panel shows the unfolded distribution for the training on events with exactly 2 jets. Instead we can train the cINN on the full inclusive dataset and evaluate it afterwards separately for the individual exclusive channels. The results of this study are displayed in the right panel of Fig. 3 and show an excellent agreement between truth distribution and unfolded data.

5. Outlook

Precision requirements of the upcoming LHC Runs will challenge current event simulation and analysis tools. At the same time, first-principles simulations are more important than ever to extract the relevant physics from the vast LHC dataset. New machine learning methods, like dedicated generative networks for precision simulations, can enhance the efficiency of established methods and open up new ways to analyze data. Multiple applications have already demonstrated that normalizing flows can provide the control that is necessary for precision simulations. We have established a comprehensive framework to estimate different types of network related uncertainties, which will be essential for any LHC application from optimized importance sampling to event generation and the inversion of the simulation chain. Given these recent developments, we expect normalizing flows to become the work horses in modern LHC simulations and their conditional counterparts to transform LHC analyses related to unfolding and optimal data exploitation.

Acknowledgments
Anja Butter is supported by the Deutsche Forschungsgemeinschaft under grant 396021762 – TRR 257 Particle Physics Phenomenology after the Higgs Discovery.

References
[1] Butter A *et al.* 2022 Machine Learning and LHC Event Generation *2022 Snowmass Summer Study* (*Preprint* 2203.07460)
[2] Ball R D *et al.* (NNPDF) 2021 *Eur. Phys. J. C* **81** 958 (*Preprint* 2109.02671)
[3] Aylett-Bullock J, Badger S and Moodie R 2021 *JHEP* **08** 066 (*Preprint* 2106.09474)
[4] Maître D and Truong H 2021 *JHEP* **11** 066 (*Preprint* 2107.06625)
[5] Danziger K, Janßen T, Schumann S and Siegert F 2021 (*Preprint* 2109.11964)
[6] Butter A and Plehn T 2020 (*Preprint* 2008.08558)
[7] Butter A, Plehn T and Winterhalder R 2019 *SciPost Phys.* **7** 075 (*Preprint* 1907.03764)
[8] Butter A, Heimel T, Hummerich S, Krebs T, Plehn T, Rousselot A and Vent S 2021 (*Preprint* 2110.13632)
[9] Otten S, Caron S, de Swart W, van Beekveld M, Hendriks L, van Leeuwen C, Podareanu D, Ruiz de Austri R and Verheyen R 2021 *Nature Commun.* **12** 2985 (*Preprint* 1901.00875)
[10] Hashemi B, Amin N, Datta K, Olivito D and Pierini M 2019 (*Preprint* 1901.05282)
[11] Di Sipio R, Faucci Giannelli M, Ketabchi Haghighat S and Palazzo S 2020 *JHEP* **08** 110 (*Preprint* 1903.02433)
[12] Arjona Martínez J, Nguyen T Q, Pierini M, Spiropulu M and Vlimant J R 2020 *J. Phys. Conf. Ser.* **1525** 012081 (*Preprint* 1912.02748)
[13] Alanazi Y, Sato N, Liu T, Melnitchouk W, Kuchera M P, Pritchard E, Robertson M, Strauss R, Velasco L and Li Y 2020 (*Preprint* 2001.11103)
[14] Howard J N, Mandt S, Whiteson D and Yang Y 2021 (*Preprint* 2101.08944)
[15] Klimek M D and Perelstein M 2020 *SciPost Phys.* **9** 053 (*Preprint* 1810.11509)
[16] Bothmann E, Janßen T, Knobbe M, Schmale T and Schumann S 2020 *SciPost Phys.* **8** 069 (*Preprint* 2001.05478)
[17] Gao C, Isaacson J and Krause C 2020 *Mach. Learn. Sci. Tech.* **1** 045023 (*Preprint* 2001.05486)
[18] Gao C, Höche S, Isaacson J, Krause C and Schulz H 2020 *Phys. Rev. D* **101** 076002 (*Preprint* 2001.10028)
[19] Chen I K, Klimek M D and Perelstein M 2021 *SciPost Phys.* **10** 023 (*Preprint* 2009.07819)
[20] Pina-Otey S, Gaitan V, Sánchez F and Lux T 2020 *Phys. Rev. D* **102** 013003 (*Preprint* 2005.12719)
[21] Butter A, Diefenbacher S, Kasieczka G, Nachman B and Plehn T 2021 *SciPost Phys.* **10** 139 (*Preprint* 2008.06545)
[22] Bieringer S, Butter A, Diefenbacher S, Eren E, Gaede F, Hundhausen D, Kasieczka G, Nachman B, Plehn T and Trabs M 2022 (*Preprint* 2202.07352)
[23] Butter A, Diefenbacher S, Kasieczka G, Nachman B, Plehn T, Shih D and Winterhalder R 2022 (*Preprint* 2202.09375)
[24] Paganini M, de Oliveira L and Nachman B 2018 *Phys. Rev. Lett.* **120** 042003 (*Preprint* 1705.02355)
[25] Krause C and Shih D 2021 (*Preprint* 2106.05285)
[26] Krause C and Shih D 2021 (*Preprint* 2110.11377)
[27] Datta K, Kar D and Roy D 2018 (*Preprint* 1806.00433)
[28] Bellagente M, Butter A, Kasieczka G, Plehn T and Winterhalder R 2020 *SciPost Phys.* **8** 070 (*Preprint* 1912.00477)
[29] Bellagente M, Butter A, Kasieczka G, Plehn T, Rousselot A, Winterhalder R, Ardizzone L and Köthe U 2020 *SciPost Phys.* **9** 074 (*Preprint* 2006.06685)
[30] Kingma D P and Welling M 2014 (*Preprint* 1312.6114)
[31] Goodfellow I J, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A and Bengio Y 2014 (*Preprint* 1406.2661)
[32] Kobyzev I, Prince S and Brubaker M 2020 *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1–1 ISSN 1939-3539 URL http://dx.doi.org/10.1109/TPAMI.2020.2992934

[33] Ardizzone L, Kruse J, Wirkert S, Rahner D, Pellegrini E W, Klessen R S, Maier-Hein L, Rother C and Köthe U 2018 (*Preprint* 1808.04730)
[34] Ardizzone L, Lüth C, Kruse J, Rother C and Köthe U 2019 (*Preprint* 1907.02392)
[35] Bollweg S, Haußmann M, Kasieczka G, Luchmann M, Plehn T and Thompson J 2020 *SciPost Phys.* **8** 006 (*Preprint* 1904.10004)
[36] Kasieczka G, Luchmann M, Otterpohl F and Plehn T 2020 *SciPost Phys.* **9** 089 (*Preprint* 2003.11099)
[37] Bellagente M, Haußmann M, Luchmann M and Plehn T 2021 (*Preprint* 2104.04543)
[38] Backes M, Butter A, Plehn T and Winterhalder R 2021 *SciPost Phys.* **10** 089 (*Preprint* 2012.07873)
[39] Stienen B and Verheyen R 2021 *SciPost Phys.* **10** 038 (*Preprint* 2011.13445)