

Generative Models in Event Simulation

Anja Butter^{a,*}

^a*Institut für Theoretische Physik,
Universität Heidelberg, Germany*

E-mail: butter@thphys.uni-heidelberg.de

First-principle based simulations are a fundamental building block of LHC physics. In the coming LHC runs, simulations will face unprecedented precision requirements, challenging the efficiency of Monte Carlo generators. With applications ranging from modular improvements to new frameworks, generative networks can help overcome these challenges.

*The Eighth Annual Conference on Large Hadron Collider Physics-LHCP2020
25-30 May, 2020
online*

*Speaker

1. Introduction

High energy particle physics owes much of its success to the impressive interplay between theory and experimental physics, which has led to first-principle based precision simulations. With the upcoming LHC Run III, and even more so for the HL-LHC, the natural limitations of computing resources pose a serious challenge to standard precision predictions. Among the main challenges are the high-dimensional phase space and intrinsic complex structures which lead to inefficient mapping of the phase space and consequently low unweighting efficiencies. In an era of data, machine learning techniques offer promising possibilities to boost the performance in order to achieve the necessary precision.

This can for instance be achieved by speeding up individual modules of the simulations via regression models that estimate amplitudes [5, 6], new phase space integration methods [7–9] or normalizing flow networks which improve standard phase space sampling [10–12]. More generally we can use generative networks to directly generate events [1, 2, 13–18]. The main advantages of generative networks include their short evaluation time and the possibility to train on simulated as well as real data. Moreover they can be used to invert a Markov Chain process, making them particularly suitable for unfolding problems [3, 19].

2. GAN training

While there are several generative networks on the market, we focus on generative adversarial networks (GANs) [20], a special type of generative networks, that is particularly well suited to generate realistic samples. A GAN consists of two networks, a generator G which transforms random noise into generated events $\{x_G\}$ and a discriminator network D , that is trained to distinguish generated events from true events x_T . Both networks are trained by minimizing their corresponding loss functions

$$L_D = \langle -\log D(x_T) \rangle_{x_T \sim P_T} + \langle -\log(1 - D(x_G)) \rangle_{x_G \sim P_G} \quad (1)$$

$$L_G = \langle -\log D(x_G) \rangle_{x_G \sim P_G} . \quad (2)$$

Once the training has converged, the generated samples follow the same underlying distribution as the true events. The stability of the training is crucial and can be improved via a variety of techniques like modified loss functions or gradient penalty.

3. Event generation

We illustrate the strengths and weaknesses of GANs using the example of $t\bar{t}$ production followed by a hadronic decay [1]. Assuming that final state particles are on-shell, the resulting phase space is 18 dimensional as long as we do not impose momentum conservation. Fig. 1 illustrates that smooth distributions like the p_T of the intermediate top quark can easily be learned, while the GAN struggles with sharper phase space features like the invariant mass of the intermediate W boson. An additional kernel loss, based on the maximum mean discrepancy of the invariant mass, can support the training of the generator, leading to a resolved mass peak. Special attention has to be applied to

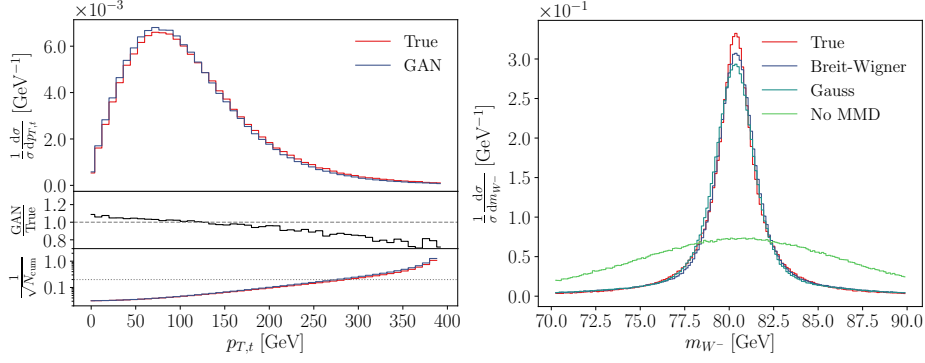


Figure 1: Distributions of the transverse momentum of the top quark (left) and the invariant mass of the W boson (right).

regions with low statistics. The relative lack of information in these regions can lead to significantly enhanced systematic uncertainties, as we can observe in the tail of the p_T distribution.

Finally we can analyze the correlations between generated observables. In Fig. 2 we show the learned correlation between the transverse momentum of the top and the bottom quark. A quantitative comparison between generated and true distribution can be achieved by slicing the parameter space and comparing the distribution of $p_{T,b}$ for fixed values of $p_{T,t}$. The right panel of Fig. 2 shows the slice with $p_{T,t} \approx 100$ GeV for which we find an excellent agreement between generated and true distribution.

Instead of only reproducing existing distributions, we can extend the GAN setup to take into account two distributions of the same dimensionality and generate their high-dimensional difference in a consistent way [2]. This allows us to include negative events into our framework. In Fig. 3 we consider the process $pp \rightarrow e^+e^-$ for which we can subtract the continuous photon background from the Z pole and the interference term. We can see how the distribution of the generated events reproduces the distribution of the training data (blue, red) as well as their difference (black). Due to the interpolation properties of the neural network, the distribution of generated events smoothly interpolates through the statistical fluctuation of the bin-wise difference.

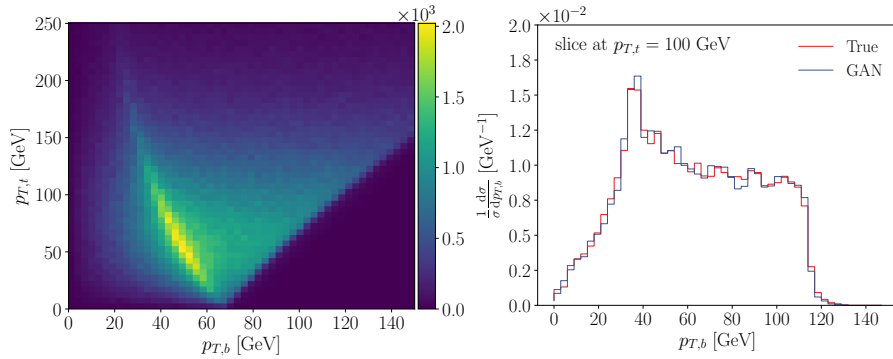


Figure 2: Generated correlation between the transverse momentum of the top and bottom quarks. The right hand side compares generated and true distribution for slice of the dataset with $p_{t,T} = 100$ GeV.

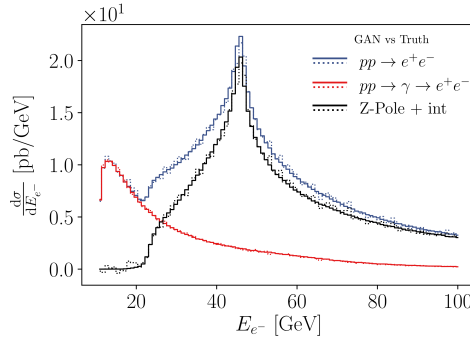


Figure 3: Difference of two distributions obtained from a bin-wise subtraction (dotted) and the subtraction-GAN (line). The distributions given by the training data are indicated in blue and red while the generated difference is highlighted in black.

4. Unfolding

While a standard GAN can learn a mapping between the parton and detector level distribution, it is not guaranteed that the resulting mapping correlates phase space regions on parton and detector level in a reasonable way. We therefore use a conditional GAN, where generator and discriminator are conditioned on the detector level information [3]. The resulting fully conditional GAN (FCGAN) can consistently unfold detector level distributions, so that phase space regions at the level of detector observables get mapped to corresponding regions of the parton level phase space. This is shown in the left panel of Fig. 4 for the process $pp \rightarrow ZW^\pm \rightarrow (l^+l^-)(jj)$, where we cut on the transverse momentum of the leading and the second leading jet. We can test the stability of the FCGAN by unfolding a new dataset, which follows a different underlying structure, but can be represented in terms of the same detector level observables. It is essential to analyze how the bias of the unfolding procedure, introduced through the training of the network, affects signs of new physics. When inserting data points with a W' resonance into the test data, we find that the position of the resonance is unfolded correctly, while the width has been slightly smeared out. While the FCGAN demonstrates a reliable performance, a statistically more sound approach can be achieved via conditional invertible networks [4].

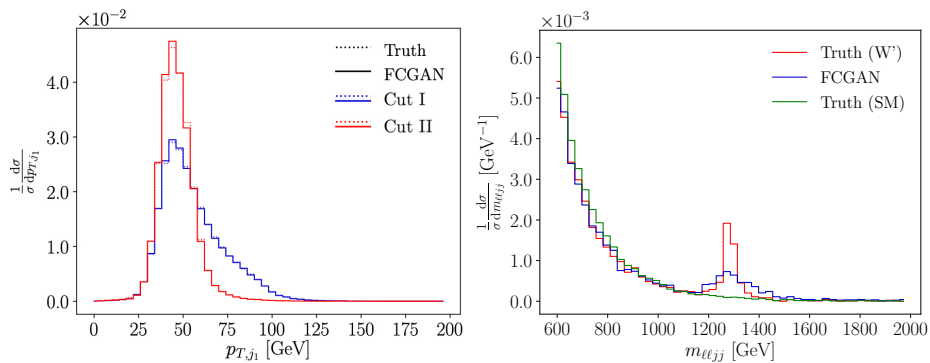


Figure 4: Parton level truth and FCGANned distributions when we train the GAN on the full data set but unfold only selected parts of phase space (left) or a new dataset including a W' resonance (right).

References

- [1] A. Butter, T. Plehn and R. Winterhalder, *SciPost Phys.* **7** (2019) no.6, 075 doi:10.21468/SciPostPhys.7.6.075 [arXiv:1907.03764 [hep-ph]].
- [2] A. Butter, T. Plehn and R. Winterhalder, [arXiv:1912.08824 [hep-ph]].
- [3] M. Bellagente, A. Butter, G. Kasieczka, T. Plehn and R. Winterhalder, *SciPost Phys.* **8** (2020) no.4, 070 doi:10.21468/SciPostPhys.8.4.070 [arXiv:1912.00477 [hep-ph]].
- [4] M. Bellagente, A. Butter, G. Kasieczka, T. Plehn, A. Rousselot, R. Winterhalder, L. Ardizzone and U. Köthe, [arXiv:2006.06685 [hep-ph]].
- [5] F. Bishara and M. Montull, [arXiv:1912.11055 [hep-ph]].
- [6] S. Badger and J. Bullock, *JHEP* **06** (2020), 114 doi:10.1007/JHEP06(2020)114 [arXiv:2002.07516 [hep-ph]].
- [7] J. Bendavid, [arXiv:1707.00028 [hep-ph]].
- [8] M. D. Klimek and M. Perelstein, [arXiv:1810.11509 [hep-ph]].
- [9] S. Carrazza and J. M. Cruz-Martinez, *Comput. Phys. Commun.* **254** (2020), 107376 doi:10.1016/j.cpc.2020.107376 [arXiv:2002.12921 [physics.comp-ph]].
- [10] E. Bothmann, T. Janßen, M. Knobbe, T. Schmale and S. Schumann, *SciPost Phys.* **8** (2020) no.4, 069 doi:10.21468/SciPostPhys.8.4.069 [arXiv:2001.05478 [hep-ph]].
- [11] C. Gao, J. Isaacson and C. Krause, doi:10.1088/2632-2153/abab62 [arXiv:2001.05486 [physics.comp-ph]].
- [12] C. Gao, S. Höche, J. Isaacson, C. Krause and H. Schulz, *Phys. Rev. D* **101** (2020) no.7, 076002 doi:10.1103/PhysRevD.101.076002 [arXiv:2001.10028 [hep-ph]].
- [13] S. Otten, S. Caron, W. de Swart, M. van Beekveld, L. Hendriks, C. van Leeuwen, D. Podareanu, R. Ruiz de Austri and R. Verheyen, [arXiv:1901.00875 [hep-ph]].
- [14] B. Hashemi, N. Amin, K. Datta, D. Olivito and M. Pierini, [arXiv:1901.05282 [hep-ex]].
- [15] R. Di Sipio, M. Fauci Giannelli, S. Ketabchi Haghighat and S. Palazzo, *JHEP* **08** (2019), 110 doi:10.1007/JHEP08(2019)110 [arXiv:1903.02433 [hep-ex]].
- [16] J. Lin, W. Bhimji and B. Nachman, *JHEP* **05** (2019), 181 doi:10.1007/JHEP05(2019)181 [arXiv:1903.02556 [hep-ph]].
- [17] J. Arjona Martínez, T. Q. Nguyen, M. Pierini, M. Spiropulu and J. R. Vlimant, *J. Phys. Conf. Ser.* **1525** (2020) no.1, 012081 doi:10.1088/1742-6596/1525/1/012081 [arXiv:1912.02748 [hep-ex]].

- [18] Y. Alanazi, N. Sato, T. Liu, W. Melnitchouk, M. P. Kuchera, E. Pritchard, M. Robertson, R. Strauss, L. Velasco and Y. Li, [arXiv:2001.11103 [hep-ph]].
- [19] K. Datta, D. Kar and D. Roy, [arXiv:1806.00433 [physics.data-an]].
- [20] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, [arXiv:1406.2661 [stat.ML]].