AMPLIFYING STATISTICS WITH ENSEMBLES OF GENERATIVE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

The potential generalization of generative models beyond their initial training data is a crucial question in many physics applications. Specifically, we ask the question whether generated samples add statistical precision beyond the training sample. For a simple toy model we show that such an amplification over the training statistics is indeed possible. We further show that ensemble methods can enhance generative performanc and measure their impact through a gain in amplification factor.

1 INTRODUCTION

Research in particle physics relies on high-fidelity simulations of fundamental processes for experiment design and statistical inference. Using a factorized Monte Carlo approach, they cover physical processes spanning nearly 20 orders of magnitude in length scales. Already now, these simulations form a major computational bottleneck. Over the coming years, the high-luminosity run of the Large Hadron Collider (LHC) will produce a 25-fold increase in data and fundamentally challenge our simulation tools. Generative machine learning models offer a way to both, speed up and improve the precision of the simulation.

There exists a landscape of generative models, including generative adversarial networks (GANs) (Goodfellow et al., 2014; Creswell et al., 2018), variational autoencoders (Kingma & Welling, 2014; 2019), and variations of normalizing flows (Rezende & Mohamed, 2015; Kobyzev et al., 2020). They can be applied to particle physics simulation, starting with phase space integration (Klimek & Perelstein, 2018; Bendavid, 2017) and phase space sampling (Bothmann et al., 2020; Gao et al., 2020b;a) including amplitude regression (Bishara & Montull, 2019; Badger & Bullock, 2020) and NN-parton showers (Bothmann & Debbio, 2019; de Oliveira et al., 2017; Monk, 2018; Andreassen et al., 2019), all the way to detector simulations (Paganini et al., 2018a;b; Musella & Pandolfi, 2018; Erdmann et al., 2018b; 2019; ATLAS Collaboration, 2018; 2019; Belayneh et al., 2020; Buhmann et al., 2020). Alternatively, events can be generated with all-in-one NN-generators (Otten et al., 2019; Hashemi et al., 2019; Di Sipio et al., 2020; Butter et al., 2019a; Alanazi et al., 2020; Howard et al., 2021), including event subtraction (Butter et al., 2019b), event unweighting Stienen & Verheyen (2021); Backes et al. (2020). In addition, generative models can improve searches for physics beyond the Standard Model (Lin et al., 2019) and searches for general anomalies (Nachman & Shih, 2020; Knapp et al., 2020). Finally, conditional GANs (Datta et al., 2018; Bellagente et al., 2020b) or flow networks (Bellagente et al., 2020a) can invert or unfold Monte Carlo simulations.

All of these applications of generative networks fall, roughly, into two categories, (i) generative networks accelerating or augmenting Monte Carlo simulations or (ii) generative networks offering new analysis opportunities. Especially the applications of the first kind raise the fundamental question of amplification, or: *How many more events can we sensibly GAN before we are limited by the statistics of the training sample*?

A seemingly straightforward answer to this question is *as many examples as were used for training, because the network does not add any physics knowledge* (Matchev & Shyamsundar, 2020). However, a key property of neural networks used in particle physics is their powerful interpolation in sparse and high-dimensional spaces. It lead to the success of the NNPDF parton densities (Del Debbio et al., 2005) as the first mainstream application of machine learning to particle physics theory.

This interpolation provides GANs with information beyond the statistical limitations of the training data.

Formally, neural networks go beyond a naive interpolation in that their architectures define basic properties of the functions it parameterizes. For example, some kind of smoothness criterion combined with a typical resolution automatically adds information to discrete training data. An extreme baseline which we will use in this submission is the case where the true density distribution is known in terms of a few unknown parameters. With this information it is always better to fit those parameters and compute statistics with the functional form than to estimate the statistics directly from the data. In the machine learning literature this kind of question is known as data amplification (Hao et al., 2019), but not extensively discussed. An interesting application is computer games, where the network traffic limits the available information while a simulation software still generates realistic images or videos. This practical question leads to more formal question of sampling correctors (Canonne et al., 2015). Alternatively, it can be linked to a classification problem to derive scaling laws for networks fooling a given hypothesis test (Hao et al., 2019; Axelrod et al., 2019).

In this submission we will study an extremely simple model, inspired by typical particle theory applications. If we know the smooth truth distribution, we can bin our space to define quantiles — intervals containing equal probability mass — and compute the χ^2-values for sampled and GANned approximations. Our example will be a camel back function because it is similar to multi-modal smooth distributions common in HEP. Compared to previous work on this problem, we also show how further improvements can be achieved by ensembles of GANs. For a more in depth discussion and additional investigation of higher dimensional data see Butter et al. (2020).

2 Problem

We use a one-dimensional camel back, made out of two normalized Gaussians $N_{\mu,\sigma}(x)$ with mean μ and width σ as our reference distribution:

$$P(x) = \frac{N_{-4,1}(x) + N_{4,1}(x)}{2} . \tag{1}$$

We show this function in Fig. 1, together with a histogrammed data set of 100 points. As a benchmark, we define a 5-parameter maximum-likelihood fit, where we assume that we know the functional form and determine the two means, the two widths and the relative height of the Gaussians in Eq. equation 1. We perform this fit using the IMINUIT (Dembinski et al., 2020) and PROBFIT (Ongmongkolkul et al., 2018) PYTHON packages. The correctly assumed functional form is much more than we can encode in a generative network, so the network will not outperform the precision of this fit. On the other hand, it is an interesting and relevant question how close a generative network can come to such a parameterized fit.

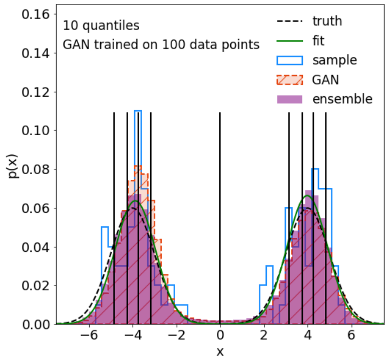

Figure 1: Camel back function as a 1D test case. We show the true distribution (black), a histogram with 100 sample points (blue), a fit to the samples data (green), a high-statistics GAN sample (orange), and a Ensemble-GAN sample (purple). Ten quantiles include 10% of the truth integral each.

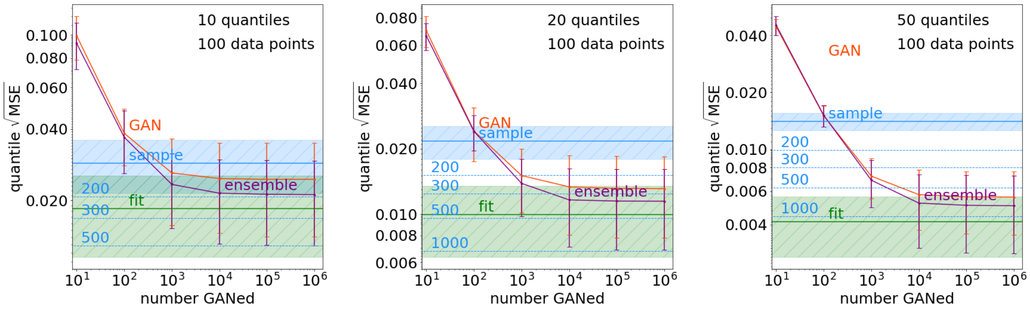

Figure 2: Quantile error for the 1D camel back function for sampling (blue), fit (green), GAN (orange), and Ensemble-GAN (purple). Left to right we show results for 10, 20, and 50 quantiles.

To quantify the agreement of all descriptions with the true distribution, we divide our phase space $x \in [-8, 8]$ into 10, 20, or 50 quantiles. The case with 10 quantiles is illustrated in Fig.1. We can then evaluate the quality of our approximation to the truth by computing the average quantile error

$$\text{MSE} = \frac{1}{N_\text{quant}} \sum_{j=1}^{N_\text{quant}} \left(p_j - \frac{1}{N_\text{quant}} \right)^2 , \tag{2}$$

where p_j is the estimated probability in each of the N_quant quantiles.In Fig. 2 the blue horizontal lines show this measure for histograms with 100 to 1000 sampled points and the fit to 100 points. For the 100-point sample, we construct an error band by evaluating 50 statistically independent samples and computing their standard deviation. For the fit, we do the same, i.e. fit the same 50 independent samples and compute the standard deviation for the fit output. This uncertainty should be equivalent to the one-sigma range of the five fitted parameters if we consider all correlations.

The first observation in Fig. 2 is that the agreement between the sample or the fit with the truth generally improves with more quantiles, which is simply a property of our quantile MSE error. Second, the fit's precision corresponds to roughly 300 hypothetical data points for 10 quantiles, 500 hypothetical data points for 20 quantiles, and close to 1000 hypothetical data points for 50 quantiles. Therefore, for high resolution and an extremely sparsely populated 1D-phase space, the assumed functional value for the fit gives this description the same statistical power as a ten times bigger dataset with no knowledge of the functional form. If we define the *amplification factor* as the ratio between asymptotic performance and training events, the amplification factor of our fit comes out around ten. The question is, how much is a GAN with its very basic assumptions worth, for instance, in comparison to this fit?

3 Model

Our generative model is a standard GAN implemented using PyTorch (Paszke et al., 2019). Both, the generator and discriminator are fully connected networks with dropout. The input noise to the generator consists of 1000 numbers sampled uniformly form $[-1, 1]$.

To aid the discriminator in learning the underlying distribution we use a mini-batch discrimination method inspired by DeepSets (Zaheer et al., 2017; Komiske et al., 2019). At every evaluation step the discriminator input is made up of a data point $x \in \mathbb{R}^1$ and the corresponding full batch $B \in \mathbb{R}^{1,n}$, with batch size n. The setup then calculates the pointwise difference $B - x \in \mathbb{R}^{1,n}$. This is passed to an embedding function $\Phi : \mathbb{R}^{1,n} \to \mathbb{R}^{m,n}$, with embedding size m. The result is then aggregate along the batch direction using a function $\text{F} : \mathbb{R}^{m,n} \to \mathbb{R}^m$. The input to the actual FCN network consist of a combination of x and $F(\Phi(B - x))$.

The discriminator and generator are trained for 10,000 epochs using the Adam optimizer (Kingma & Ba, 2014) with an exponential learning rate decay. The discriminator training is improved using a gradient penalty (Roth et al., 2017), which significantly stabilizes the training process. To further prevent the network from overfitting to the small dataset, we modify each real data point with linear noise drawn from $[-0.1, 0.1]$. This noise is re-sampled at every evaluation step.

To further alleviate training instabilities, we introduce an additional setup we term Ensemble-GAN. For this, we train N independent GANs on each 100 data point training sample as described above. We then sample $1/N$ of the total GANed samples from each of these models during the generation process. In our setup we choose $N = 10$.

4 Results

Returning to Fig. 2, the orange and purple curves show the performance of the GAN and Ensemble-GAN setup compared to the sample and the 5-parameter fit. The uncertainties for both are again derived via 50 independent training repetitions. We compute the quantile error as a function of the number of GANned events and see how it saturates. This saturation is where GANning more events would not add more information to a given sample. Depending on the number of quantiles or the resolution, the information saturates between 1000 and 100,000 GANned events, compared with 100 training events. This result shows that it does make sense to generate more events than the training sample size.

On the other hand, a GANned event does not carry the same amount of information as a sampled event. The asymptotic value of the GAN quantile error should be compared to the expected quantile error for an increased sample, and we find that the 10,000 GANned events are worth between 150 and 600 sampled events, depending on the applied resolution. The ensemble method pushes this number of corresponding sampled events to 190 and 750, indicating that it indeed improves the setup's convergence behaviour. Altogether, our GAN produces an amplification factor above seven for a resolution corresponding to 50 quantiles. A noteworthy feature of the GAN is that it follows the fit result with a slight degradation, roughly corresponding to the quoted fit uncertainty. In that sense, the analogy between a fit and a flexible NN-representation makes sense, and the GAN comes shockingly close to the highly constrained fit. This is despite the GAN being significantly more versatile and complex and therefore more prone to over-fitting than the parameter fit. Finally, just like a parameterized fit, training a GAN encodes statistical uncertainties in the training sample into systematic uncertainties in the network.

5 Outlook

A crucial question for applications of generative models to particle physics simulations is how many events we can generate through the fast network before we exhaust the physics information encoded in the training sample. Generally, a neural network adds information or physics knowledge through the class of functions it represents. To answer this question for GANs, we split the phase space of a simple test function into quantiles and use the combined quantile MSE to compare sample, GAN, and a fit benchmark.

We find that it makes sense to GAN significantly more events than we have in the training sample, but those individual events carry less information than a training sample event. As GAN sampling can be much more computationally efficient than ab initio simulations, this triggers a net statistical benefit. We define an *amplification factor* as the number of hypothetical training events with the same information content as asymptotically many GANned events. This amplification factor strongly depends on the number of quantiles or the phase space resolution. While we can never beat a low-parameter fit, the GAN comes surprisingly close, and its amplification factor scales with the fit's amplification factor.

While our focus is on GANs, our observed amplification is also relevant for other neural networks applications. Specifically, it also applies to networks increasing the data quality through refining (Erdmann et al., 2018a; Bellagente et al., 2020b) or reweighting (Andreassen & Nachman, 2020; Andreassen et al., 2020). It will be interesting to see how the improvement scales with the amount of training data for these approaches.

REFERENCES

Yasir Alanazi, N. Sato, Tianbo Liu, W. Melnitchouk, Michelle P. Kuchera, Evan Pritchard, Michael Robertson, Ryan Strauss, Luisa Velasco, and Yaohang Li. Simulation of electron-proton scattering events by a Feature-Augmented and Transformed Generative Adversarial Network (FAT-GAN). 1 2020. 2001.11103.

Anders Andreassen and Benjamin Nachman. Neural Networks for Full Phase-space Reweighting and Parameter Tuning. *Phys. Rev. D*, 101(9):091901, 2020. doi: 10.1103/PhysRevD.101.091901.

Anders Andreassen, Ilya Feige, Christopher Frye, and Matthew D. Schwartz. JUNIPR: a Framework for Unsupervised Machine Learning in Particle Physics. *Eur. Phys. J.*, C79(2):102, 2019. doi: 10.1140/epjc/s10052-019-6607-9.

Anders Andreassen, Patrick T. Komiske, Eric M. Metodiev, Benjamin Nachman, and Jesse Thaler. OmniFold: A Method to Simultaneously Unfold All Observables. *Phys. Rev. Lett.*, 124(18): 182001, 2020. doi: 10.1103/PhysRevLett.124.182001.

ATLAS Collaboration. Deep generative models for fast shower simulation in ATLAS. ATL-SOFT-PUB-2018-001, 2018. http://cds.cern.ch/record/2630433.

ATLAS Collaboration. Energy resolution with a GAN for Fast Shower Simulation in ATLAS. ATLAS-SIM-2019-004, 2019. https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PLOTS/SIM-2019-004/.

Brian Axelrod, Shivam Garg, Vatsal Sharan, and Gregory Valiant. Sample amplification: Increasing dataset size even when learning is impossible. 2019. 1904.12053.

Mathias Backes, Anja Butter, Tilman Plehn, and Ramon Winterhalder. How to GAN Event Unweighting. 12 2020. 2012.07873.

Simon Badger and Joseph Bullock. Using neural networks for efficient evaluation of high multiplicity scattering amplitudes. 2 2020. 2002.07516.

Dawit Belayneh et al. Calorimetry with Deep Learning: Particle Simulation and Reconstruction for Collider Physics. *Eur. Phys. J. C*, 80(7):688, 2020. doi: 10.1140/epjc/s10052-020-8251-9.

Marco Bellagente, Anja Butter, Gregor Kasieczka, Tilman Plehn, Armand Rousselot, Ramon Winterhalder, Lynton Ardizzone, and Ullrich Köthe. Invertible Networks or Partons to Detector and Back Again. 6 2020a. 2006.06685.

Marco Bellagente, Anja Butter, Gregor Kasieczka, Tilman Plehn, and Ramon Winterhalder. How to GAN away Detector Effects. *SciPost Phys.*, 8(4):070, 2020b. doi: 10.21468/SciPostPhys.8.4.070.

Joshua Bendavid. Efficient Monte Carlo Integration Using Boosted Decision Trees and Generative Deep Neural Networks. 2017. 1707.00028.

Fady Bishara and Marc Montull. (Machine) Learning Amplitudes for Faster Event Generation. 12 2019. 1912.11055.

Enrico Bothmann and Luigi Debbio. Reweighting a parton shower using a neural network: the final-state case. *JHEP*, 01:033, 2019. doi: 10.1007/JHEP01(2019)033.

Enrico Bothmann, Timo Janßen, Max Knobbe, Tobias Schmale, and Steffen Schumann. Exploring phase space with Neural Importance Sampling. *SciPost Phys.*, 8(4):069, 2020. doi: 10.21468/SciPostPhys.8.4.069.

Erik Buhmann, Sascha Diefenbacher, Engin Eren, Frank Gaede, Gregor Kasieczka, Anatolii Korol, and Katja Krüger. Getting High: High Fidelity Simulation of High Granularity Calorimeters with High Speed. 5 2020. 2005.05334.

Anja Butter, Tilman Plehn, and Ramon Winterhalder. How to GAN LHC Events. *SciPost Phys.*, 7: 075, 2019a. doi: 10.21468/SciPostPhys.7.6.075.

Anja Butter, Tilman Plehn, and Ramon Winterhalder. How to GAN Event Subtraction. 2019b. 1912.08824.

Anja Butter, Sascha Diefenbacher, Gregor Kasieczka, Benjamin Nachman, and Tilman Plehn. GANplifying Event Samples. 8 2020. 2008.06545.

Clément Canonne, Themis Gouleakis, and Ronitt Rubinfeld. Sampling correctors. 2015. 1504.06544.

Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A. Bharath. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35 (1):53–65, Jan 2018. ISSN 1053-5888. doi: 10.1109/msp.2017.2765202. URL http://dx.doi.org/10.1109/MSP.2017.2765202.

Kaustuv Datta, Deepak Kar, and Debarati Roy. Unfolding with Generative Adversarial Networks. 2018. 1806.00433.

Luke de Oliveira, Michela Paganini, and Benjamin Nachman. Learning Particle Physics by Example: Location-Aware Generative Adversarial Networks for Physics Synthesis. *Comput. Softw. Big Sci.*, 1(1):4, 2017. doi: 10.1007/s41781-017-0004-6.

Luigi Del Debbio, Stefano Forte, Jose I. Latorre, Andrea Piccione, and Joan Rojo. Unbiased determination of the proton structure function F(2)**p with faithful uncertainty estimation. *JHEP*, 03: 080, 2005. doi: 10.1088/1126-6708/2005/03/080.

Hans Dembinski, Piti Ongmongkolkul, Christoph Deil, David Menéndez Hurtado, Matthew Feickert, Henry Schreiner, Andrew, Chris Burr, Fabian Rost, Alex Pearce, Lukas Geiger, Bernhard M. Wiedemann, and Omar Zapata. scikit-hep/iminuit: v1.4.9, July 2020. URL https://doi.org/10.5281/zenodo.3951328.

Riccardo Di Sipio, Michele Faucci Giannelli, Sana Ketabchi Haghighat, and Serena Palazzo. DijetGAN: A Generative-Adversarial Network Approach for the Simulation of QCD Dijet Events at the LHC. *JHEP*, 08:110, 2020. doi: 10.1007/JHEP08(2019)110.

Martin Erdmann, Lukas Geiger, Jonas Glombitza, and David Schmidt. Generating and refining particle detector simulations using the Wasserstein distance in adversarial networks. *Comput. Softw. Big Sci.*, 2(1):4, 2018a. doi: 10.1007/s41781-018-0008-x.

Martin Erdmann, Lukas Geiger, Jonas Glombitza, and David Schmidt. Generating and refining particle detector simulations using the Wasserstein distance in adversarial networks. *Comput. Softw. Big Sci.*, 2(1):4, 2018b. doi: 10.1007/s41781-018-0008-x.

Martin Erdmann, Jonas Glombitza, and Thorben Quast. Precise simulation of electromagnetic calorimeter showers using a Wasserstein Generative Adversarial Network. *Comput. Softw. Big Sci.*, 3:4, 2019. doi: 10.1007/s41781-018-0019-7.

Christina Gao, Stefan Höche, Joshua Isaacson, Claudius Krause, and Holger Schulz. Event Generation with Normalizing Flows. *Phys. Rev. D*, 101(7):076002, 2020a. doi: 10.1103/PhysRevD.101.076002.

Christina Gao, Joshua Isaacson, and Claudius Krause. i-flow: High-dimensional Integration and Sampling with Normalizing Flows. 2020b. 2001.05486.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pp. 2672–2680, Cambridge, MA, USA, 2014. MIT Press. URL http://dl.acm.org/citation.cfm?id=2969033.2969125.

Yi Hao, Alon Orlitsky, Ananda T. Suresh, and Yihong Wu. Data amplification: A unified and competitive approach to property estimation. 2019. 1904.00070.

Bobak Hashemi, Nick Amin, Kaustuv Datta, Dominick Olivito, and Maurizio Pierini. LHC analysis-specific datasets with Generative Adversarial Networks. 2019. 1901.05282.

Jessica N. Howard, Stephan Mandt, Daniel Whiteson, and Yibo Yang. Foundations of a Fast, Data-Driven, Machine-Learned Simulator. 1 2021. 2101.08944.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. 2014. 1412.6980.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun (eds.), *ICLR*, 2014. URL http://dblp.uni-trier.de/db/conf/iclr/iclr2014.html#KingmaW13.

Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019. ISSN 1935-8245. doi: 10.1561/2200000056. URL http://dx.doi.org/10.1561/2200000056.

Matthew D. Klimek and Maxim Perelstein. Neural Network-Based Approach to Phase Space Integration. 2018. 1810.11509.

Oliver Knapp, Guenther Dissertori, Olmo Cerri, Thong Q. Nguyen, Jean-Roch Vlimant, and Maurizio Pierini. Adversarially Learned Anomaly Detection on CMS Open Data: re-discovering the top quark. 5 2020. 2005.01598.

Ivan Kobyzev, Simon Prince, and Marcus Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020. ISSN 1939-3539. doi: 10.1109/tpami.2020.2992934. URL http://dx.doi.org/10.1109/TPAMI.2020.2992934.

Patrick T. Komiske, Eric M. Metodiev, and Jesse Thaler. Energy Flow Networks: Deep Sets for Particle Jets. *JHEP*, 01:121, 2019. doi: 10.1007/JHEP01(2019)121.

Joshua Lin, Wahid Bhimji, and Benjamin Nachman. Machine Learning Templates for QCD Factorization in the Search for Physics Beyond the Standard Model. *JHEP*, 05:181, 2019. doi: 10.1007/JHEP05(2019)181.

Konstantin T. Matchev and Prasanth Shyamsundar. Uncertainties associated with GAN-generated datasets in high energy physics. 2 2020. 2002.06307.

J. W. Monk. Deep Learning as a Parton Shower. *JHEP*, 12:021, 2018. doi: 10.1007/JHEP12(2018)021.

Pasquale Musella and Francesco Pandolfi. Fast and Accurate Simulation of Particle Detectors Using Generative Adversarial Networks. *Comput. Softw. Big Sci.*, 2(1):8, 2018. doi: 10.1007/s41781-018-0015-y.

Benjamin Nachman and David Shih. Anomaly Detection with Density Estimation. *Phys. Rev. D*, 101:075042, 2020. doi: 10.1103/PhysRevD.101.075042.

Piti Ongmongkolkul, Christoph Deil, Chih hsiang Cheng, Alex Pearce, Eduardo Rodrigues, Henry Schreiner, Matthieu Marinangeli, Lukas Geiger, and Hans Dembinski. scikit-hep/probfit: 1.1.0, November 2018. URL https://doi.org/10.5281/zenodo.1477853.

Sydney Otten et al. Event Generation and Statistical Sampling with Deep Generative Models and a Density Information Buffer. 2019. 1901.00875.

Michela Paganini, Luke de Oliveira, and Benjamin Nachman. Accelerating Science with Generative Adversarial Networks: An Application to 3D Particle Showers in Multilayer Calorimeters. *Phys. Rev. Lett.*, 120(4):042003, 2018a. doi: 10.1103/PhysRevLett.120.042003.

Michela Paganini, Luke de Oliveira, and Benjamin Nachman. CaloGAN : Simulating 3D high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks. *Phys. Rev.*, D97(1):014021, 2018b. doi: 10.1103/PhysRevD.97.014021.

Adam Paszke et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems 32*, pp. 8024–8035, 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pp. 1530–1538. JMLR.org, 2015.

Kevin Roth, Aurélien Lucchi, Sebastian Nowozin, and Thomas Hofmann. Stabilizing training of generative adversarial networks through regularization. *Advances in Neural Information Processing Systems (NIPS)*, 2017. 1705.09367. URL http://arxiv.org/abs/1705.09367.

Bob Stienen and Rob Verheyen. Phase Space Sampling and Inference from Weighted Events with Autoregressive Flows. *SciPost Phys.*, 10:038, 2021. doi: 10.21468/SciPostPhys.10.2.038.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander Smola. Deep sets. 2017. 1703.06114.