EPJ Web of Conferences **245**, 02034 (2020) https://doi.org/10.1051/epjconf/202024502034
CHEP 2019

Fast simulation of electromagnetic particle showers in high granularity calorimeters

Ricardo Brito Da Rocha[1,*], *Federico* Carminati[1,**], *Gulrukh* Khattak[1,***], and *Sofia* Vallecorsa[1,****]

[1]CERN, 1 Esplanade des Particules, Geneva, Switzerland

Abstract. The future need of simulated events by the LHC experiments and their High Luminosity upgrades, is expected to increase by one or two orders of magnitude. As a consequence, research on new fast simulation solutions, including deep Generative Models, is very active and initial results look promising.
We have previously reported on a prototype that we have developed, based on 3 dimensional convolutional Generative Adversarial Network, to simulate particle showers in high-granularity calorimeters. In this contribution we present improved results on a more realistic simulation. Detailed validation studies show very good agreement with Monte Carlo simulation. In particular, we show how increasing the network representational power, introducing physics-based constraints and using a transfer-learning approach for training improve the level of agreement over a large energy range.

1 Introduction

High Energy Physics (HEP) relies heavily on Monte Carlo simulation in order to model complex processes and describe detector response. The classical Monte Carlo approach can reproduce theoretical expectations with a high level of precision but it is both time and resource intensive [1]. Existing fast simulation techniques are mostly based on parametrization [2–4] or look up table like approaches [5] providing different levels of accuracy. Deep Neural Networks are also being investigated as fast simulation alternative solutions [6–9]. Calorimeters are among the most time consuming detectors as far as simulation is concerned. Their output can be regarded as a pattern of energy depositions that can be interpreted as pixel intensities in an image. These 3D images are characteristic of the particle type, its energy and the incident angle with respect to the orthogonal to the calorimeter cell face. Hence, these variables are input to the simulation process. The work presented in [9] demonstrated the benefits of using the primary particle energy to condition the training of a DNN. Here we introduce a more realistic use case by conditioning our model on the incident angle as well: the network thus learns a joint distribution of both primary energy and incident angle. With respect to [9], this model introduces new features to reach a higher level of accuracy: in particular we found that domain knowledge is useful for tuning the optimization procedure.

*e-mail: ricardo.Rocha@cern.ch
**e-mail: Federico.Carminati@cern.ch
***e-mail: gul.rukh.khattak@cern.ch
****e-mail: sofia.vallecorsa@cern.ch

1.1 Previous Work

Generative Adversarial Networks (GAN)[10] implement the idea of adversarial training using two neural networks: a generator that reproduces the true data distribution and a discriminator, typically a classifier discriminating generated from true examples. Training develops as a minimax optimisation process reaching, ideally, a saddle point that corresponds to a minimum for the generator and a maximum for the discriminator (Nash equilibrium). The Auxiliary Classifier GAN (ACGAN) follows a semi-supervised approach and demonstrates that the introduction of a label results in faster convergence and stable performance [11]. There have been many recent variations of GAN: our work combines an ACGAN-like approach with physics-derived constraints. The LAGAN [12] and CaloGAN [8] models represent the first applications of GAN to HEP simulation: particle showers in simplified calorimeters are simulated as a set of two-dimensional images. Further examples are described in [6, 7]. Simulation of highly granular calorimeters using true 3D convolutions to fully exploit the correlations in the volumetric space achieves promising results [9]. Additional examples of Generative Models applied to the simulation of detector output can be found in different contributions of these proceedings.

2 The 3D convolutional GAN

In the present work we focus on the simulation of a high granularity electromagnetic calorimeter (ECAL), designed in the context of the detector studies for the CLIC accelerator project [13]. It consists of a regular grid of $5.1\,mm^3$ cells with an inner calorimeter radius of $1.5\,mm$. The corresponding data was generated in an effort to provide a common realistic data set that could be used to foster development of different Deep Learning and Machine Learning applications [14]. Data samples were generated using a detailed Monte Carlo approach (with the Geant4 toolkit [15]). Here, we show results obtained using $400,000$ single electrons with a primary energy (E_p) range of $2 - 500$ GeV and incident angle (θ) uniformly distributed between 60° and 120°. This data are pre-processed to generate three-dimensional $51 \times 51 \times 25$ pixelized images centered around the barycenter of the energy depositions. 2D projections for an example event are shown in shown in Figure 1.

The 3DGAN model, described in [9], represents a first proof of concept of the possibility to use 3D convolutional GANs to simulate high granularity calorimeters. This work extends the scope of [9] and generates more realistic simulations with particles entering the detector with a variable incident angle and generating images that are 4 times larger. The training is conditioned using both the incident angle and energy of the incoming particle, therefore, 3DGAN learns an angle-energy multivariate distribution. The networks architecture and the corresponding loss functions are modified to take into account these new features and introduce physics-based constraints. Figure 2 shows the 3DGAN architecture. The generator input is obtained by concatenating a latent vector of 254 random numbers drawn from a Gaussian distribution to the primary particle energy and the incident angle. The next step is represented by a set of up-sampling operations that are clustered together before any convolution is applied. A faithful representation of the energy shower vs. incident angle distribution is obtained by a setup in which the generator is stronger (seven convolutional layers) than the discriminator (four layers). Further details on the networks architecture can be found in [16]. The discriminator output is two-fold: a sigmoid neuron predicts the typical GAN real/fake probability and a linear neuron implements a regression on the primary particle energy. In addition, the total deposited energy, a binned pixel intensity distribution, and the incident angle are calculated from the images using lambda functions and constrained at training time.

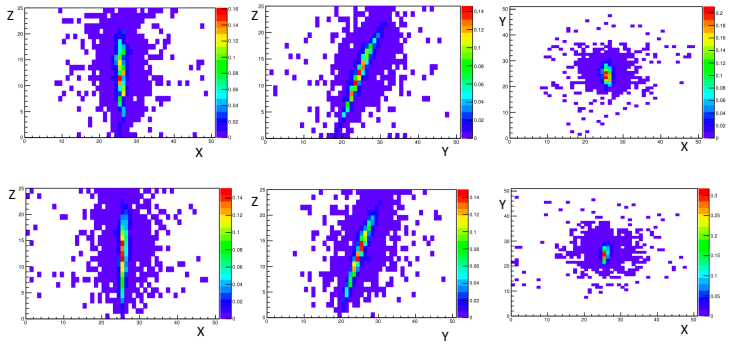

Figure 1. Example 2D energy shower sections on the xz, yz and xy planes for primary particle energy and incident angle equal to 142.52 GeV and 63.79°. Top: Geant4 electron. Bottom: GAN electron. The z axis lies along the detector depth and x, y are the transverse axes.

Given the large pixel dynamic range, shown in Figure 3, we do not apply the standard normalization-rescaling procedure. In order to slightly reduce the pixel dynamic range, we calculate the power function of pixels intensities using an exponent smaller than one. The exponent is treated as a hyper-parameter and adjusted, to a value of 0.85, through a trial-and-error procedure. The training process is split into two steps: initially, we restrict the primary particle energy range to $100 - 200$ GeV to reduce sample variability, we then extend the energy to the full $2 - 500$ GeV range, using a transfer learning approach. The discriminator and the generator are trained alternatively for the same number of steps.The architecture is implemented using keras 1.2.2 [17] (and Tensorflow 1.0.0 [18] as a backend). Training on a single NVIDIA GeForce GTX 1080 card for one epoch requires about two hours and convergence is reached after 60 epochs.

3 Results and Discussion

We validate the 3DGAN results performing a detailed comparison to Monte Carlo simulated data. Figures 1, 4 and 5 present results obtained by training the network on the full energy range. Figure 1 presents an example of 2D sections of the energy showers corresponding to particles entering the calorimeter with the same incident angles and energies for both Geant4 and GAN: the corresponding Figure 4 presents the energy shower profiles along the x, y and z axes for different angles, in both linear (for 90° incident angle) and log scale. The agreement is very good: the network is capable of correctly reproducing the spatial distribution of energy deposits as a function of the incident angle, across a large dynamic range. The largest discrepancies appear at the edges of the simulated volumes, where 3DGAN predicts, on average, smaller energy depositions. It should be noted, however, that the amount of energy expected in this regions is very small (well below 10^{-4} GeV). We obtain a similar agreement in terms of the sampling fraction in Figure 5: the network correctly reproduces the Geant4 behaviour over the entire energy range. The results obtained by training the optimized architecture on the full energy range, (from 2 to 500 GeV), for seven additional epochs on a larger sample ($400,000$ events): 3DGAN successfully generalizes the results obtained on the smaller range to the larger one. Figure 3, shows the good level of agreement on the the

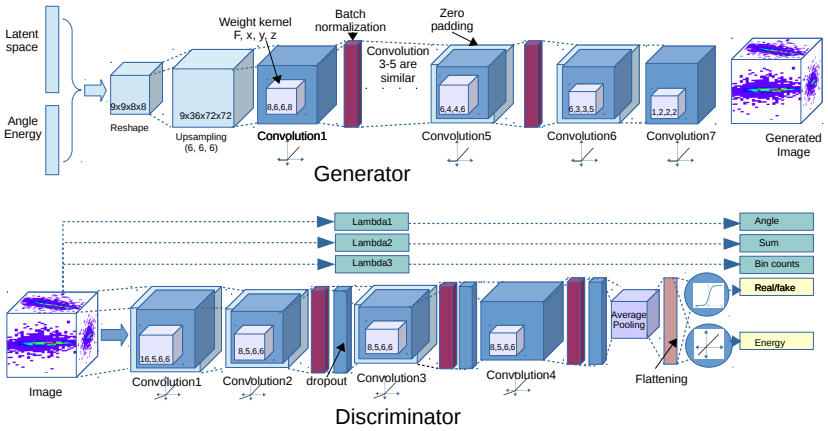

Figure 2. The 3DGAN architecture

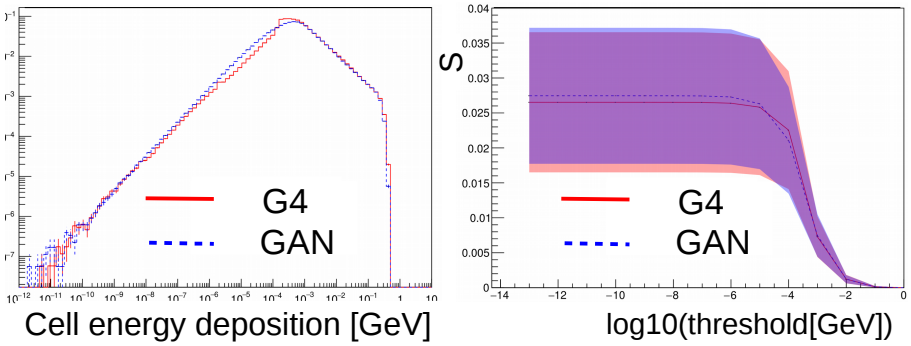

Figure 3. GAN vs, Geant4 for 2-500 GeV. Left: individual cell energies. Right: fraction of cells above a threshold for different values of threshold.

pixel intensities (left) and the sparsity (right). It should be noted (Figure 3 left panel) that the network smooths out the features present in the original Geant4 spectrum.

4 Automatic deployment of 3DGAN distributed training on Cloud

In general terms, simulating samples using generative models is much faster than using a Monte Carlo approach. In our case we observe several orders of magnitude of speed-up. We have measured the time to simulate a single electron shower in about 2 milliseconds running 3DGAN on a NVIDIA GeForce 1080 GPU [1].

The training process is however very time consuming: an entire week is needed in order to train the model to convergence using a single gaming GPU, such as the NVIDIA GTX model

[1] training time is about 17 seconds running Geant4 on a Intel Xeon 8180 (currently it is not possible to run a Geant4-based simulation on GPUs)

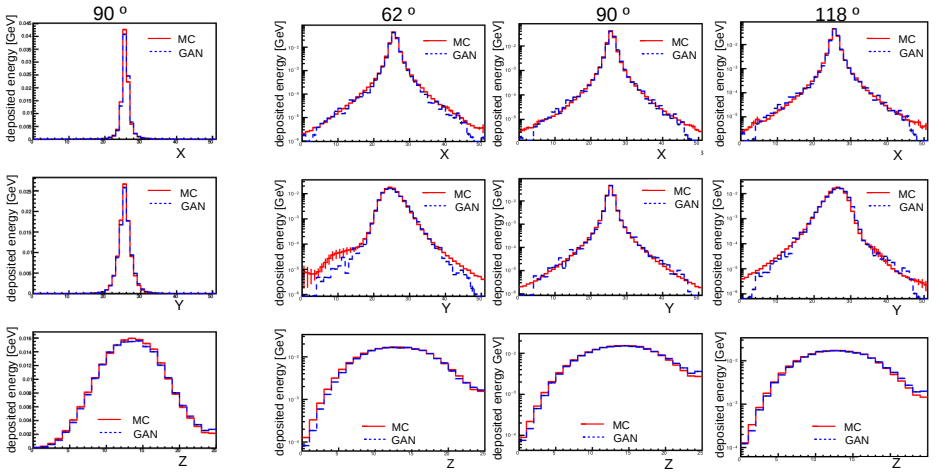

Figure 4. Shower shapes for Geant 4 vs. GAN events along x, y and z axis:left) for 90° incident angle, in linear scale; right) for 62°, 90° and 118°, in log scale.

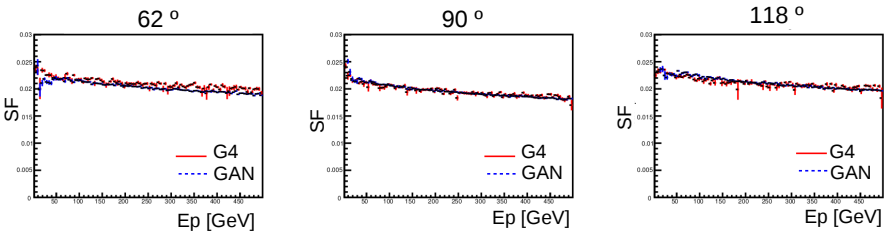

Figure 5. Sampling fraction for GAN and Geant4 events for three different angle values.

and therefore a distributed training approach is essential [19]. In order to reduce the training time we have interfaced 3DGAN to several distributed frameworks, including Horovod [20] and mpi-learn [21], and benchmarked the parallel training process on different HPC systems [19, 22].

Here we report on updated results on 3DGAN scaling performance on public clouds. Several initiatives exist that aim at understanding how the scientific community can integrate public clouds in their computing models. The European Commission funded project Helix Nebula Science Cloud (HNSciCloud) [23], for example, explored an hybrid cloud model linking together commercial cloud service providers and research organisations' in-house resources in order to provide an innovative hybrid architecture to support the growing computing needs of the research community. We have created a mpi-learn based docker [24] image and integrated it to kubernetes [25] and kubeflow [26] in order to smoothly deploy our workload on commercial cloud providers, via the HNSciCloud project. Results are shown in Figure 6: we have tested different deployment configurations and no overhead, due to the docker, kubernetes or kubeflow additional layer, has been observed. We have compared results obtained using Exoscale[2] (equipped with NVIDIA P100 GPUs) (blue) to the speed-up

[2]https://www.exoscale.com

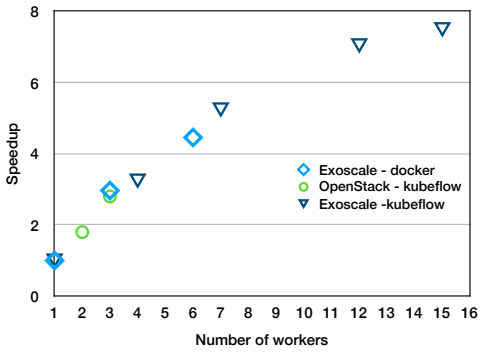

Figure 6. Speed-up for training 3DGAN, as a function of the number of MPI workers and with respect to training with one MPI worker.

measured on a small local set of GPUs, available in CERN Openstack (green) and we observed no difference in timing. Training time is significantly reduced, but the current speed-up is not linear. A possible explanation is that the workload for the workers is too small with respect to communication time and weights updates processing by the master. Analysis and optimisation of resource usage is part of our on-going work.

5 Conclusions

The 3DGAN model is capable of realistically reproducing single particle showers in high granularity calorimeters, the same kind that will be operated, in some years, at the High Luminosity LHC [27] and at next generation particle accelerators. Detailed validation studies show that the 3DGAN images reproduce the classical Monte Carlo behaviour, within just a few percents over a very large dynamical range. We intend to continue this work in order to bring the 3DGAN prototype to production-level quality, following two main R&D directions: an investigation on the generalization capabilities of the model (whether it is possible to tune the architecture parameters in order to simulate different calorimters) and a deeper study on the 3DGAN performance in terms of dataset mixing and phase space coverage, sample diversity and size of the generator support space.

References

[1] The HEP Software Foundation, J. Albrecht et al., Computing and Software for Big Science **3**, 7 (2019)
[2] W. Lucas, in *International Conference on Computing in High Energy and Nuclear Physics* (2012), Vol. 396
[3] D. Orbaker, in *International Conference on Computing in High Energy and Nuclear Physics* (2010), Vol. 219
[4] D. Autiero et al. (NOMAD Collaboration), Nucl. Instrum. Methods Phys. Res., A **425**, 188. 28 p (1998)
[5] E. Barberio et al., J. Phys. Conf. Ser. **160**, 012082 (2009)
[6] M. Erdmann et al., `https://arxiv.org/abs/1807.01954`
[7] V. Chekalina et al., `https://arxiv.org/abs/1812.01319`

[8] M. Paganini, L. de Oliveira, B. Nachman, arXiv preprint arXiv:1705.02355 (2017)
[9] G.R. Khattak, S. Vallecorsa, F. Carminati, in *2018 25th IEEE International Conference on Image Processing (ICIP)* (2018), pp. 3913–3917, ISSN 2381-8549
[10] I.J. Goodfellow et al., ArXiv e-prints (2014), 1406.2661
[11] A. Odena, C. Olah, J. Shlens, ArXiv e-prints (2016), 1610.09585
[12] L. de Oliveira, M. Paganini, B. Nachman, arXiv preprint arXiv:1701.05927 (2017)
[13] CERN, http://clic-study.web.cern.ch/
[14] F. Carminati et al., in *NIPS* (2017), https://dl4physicalsciences.github.io/files/nips_dlps_2017_15.pdf
[15] CERN, *Geant* (accessed July 31, 2017), http://geant.cern.ch/
[16] G. Khattak et al., in *IEEE International Conference on Machine Learning and Applications, ICML2019* (2019)
[17] F. Chollet et al., *Keras*, https://github.com/fchollet/keras (2015)
[18] M. Abadi et al. (2015), software available from tensorflow.org, https://www.tensorflow.org/
[19] J.R. Vlimant et al., in *CHEP 2018 conference, in publication* (2018)
[20] A. Sergeev, M.D. Balso, CoRR **abs/1802.05799** (2018), 1802.05799
[21] D. Anderson, J. Vlimant, M. Spiropulu, CoRR **abs/1712.05878** (2017), 1712.05878
[22] S. Vallecorsa et al., in *High Performance Computing* (2018), Vol. 11203, https://doi.org/10.1007/978-3-030-02465-9_35
[23] J. Fernandes et al., *Activity report HNSciCloud pilot phase* (2018)
[24] D. Merkel, Linux J. **2014** (2014)
[25] The Kubernetes authors, [Online; accessed 31-May-2019], https://kubernetes.io/
[26] The Kubeflow authors, [Online; accessed 31-May-2019], https://www.kubeflow.org/
[27] G. Apollinari et al., CERN Yellow Rep. Monogr. **4**, 1 (2017)