BENCHMARK OF GENERATIVE ADVERSARIAL NETWORKS FOR FAST HEP CALORIMETER SIMULATIONS

F. Rehm[1,2,a], S. Vallecorsa[1], K. Borras[2,3], D. Krücker[3]

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

[2] *RWTH Aachen University, Templergraben 55, Aachen, Germany*

[3] *DESY, Notkestraße 85, Hamburg, Germany*

E-mail: [a] florian.matthias.rehm@cern.ch

Highly precise simulations of elementary particles interaction and processes are fundamental to accurately reproduce and interpret the experimental results in High Energy Physics (HEP) detectors and to correctly reconstruct the particle flows. Today, detector simulations typically rely on Monte Carlo-based methods which are extremely demanding in terms of computing resources. The need for simulated data at future experiments - like the ones that will run at the High Luminosity Large Hadron Collider (HL-LHC) - are expected to increase by orders of magnitude, increasing drastically the computational challenge. This expectation motivates the research for alternative deep learning-based simulation strategies.

In this research we speed-up HEP detector simulations for the specific case of calorimeters using Generative Adversarial Networks (GANs) with a huge factor of over *150 000x* compared to the standard Monte Carlo simulations. This could only be achieved by designing smart convolutional 2D network architectures for generating 3D images representing the detector volume. Detailed physics evaluation shows an accuracy similar to the Monte Carlo simulation.

Furthermore, we quantize the data format for the neural network architecture (float32) with the Intel Low Precision Optimization tool (LPOT) to a reduced precision (int8) data format. This results in an additional *1.8x* speed-up on modern Intel hardware while maintaining the physics accuracy. These excellent results consolidate the beneficial use of GANs for future fast detector simulations.

Keywords: Generative Adversarial Networks, Calorimeter Simulation, Fast Simulation, Reduced Precision Computing

Florian Rehm, Sofia Vallecorsa, Kerstin Borras, Dirk Krücker

1. HEP Calorimeter Simulations

At present, detector simulations are primarily performed with the Geant4 toolkit [1] which relies on Monte Carlo-based methods. Calorimeters are detectors that measure the particles energy in high energy physics experiments such as at the Large Hadron Collider (LHC). Due to their considerable complexity and high granularity calorimeter simulations remain the tasks which utilize the most significant fraction of computational resources. In the future High Luminosity LHC (HL-LHC) phases the amount of data to be simulated will significantly increase due to the larger luminosities. Furthermore, the calorimeter detectors get progressively more complex with higher granularities. This predictably causes an increase of the computational requirements which exceed the extrapolated computational resources of the Worldwide LHC Computing Grid [2] by far.

In this research are Generative Adversarial Networks (GANs) - a modern Deep Learning approach - applied to speed-up calorimeter simulations. Recent physics publications proved already speed-up's of orders of magnitudes [3, 4] while maintaining physics accuracy [5, 6]. As training data are *200 000* three-dimensional high granularity shower images with a dimension of 25x25x25 pixels used. One demonstrative example shower image is shown in Figure 1.

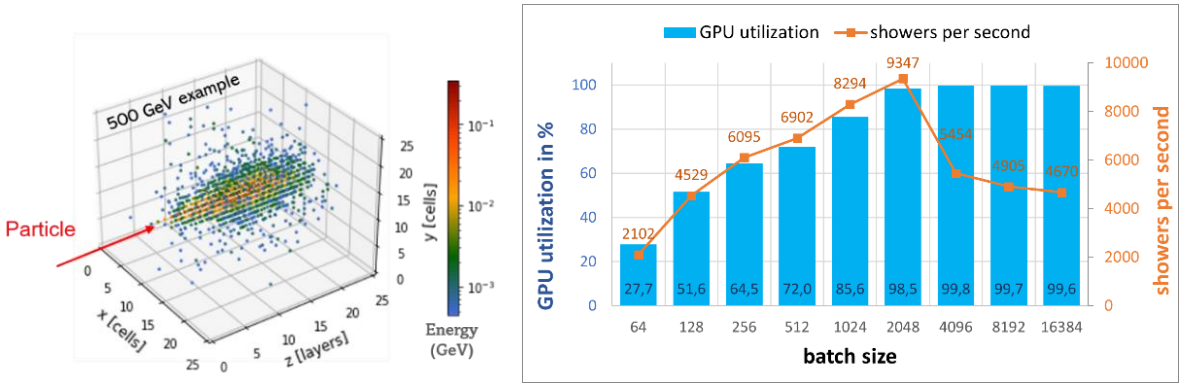

Figure 1. Shows (left) an example electromagnetic calorimeter 3D shower image with a primary particle energy of 500 GeV. (right) Inference of Conv2D model run with different batch sizes. With a batch size of 2 048 it reveals the highest inference time with 9 347 showers per second (or 158 000x speed-up versus Geant4).

2. 3D Generative Adversarial Network

Deep learning approaches are today an appropriate choice to deal with computationally demanding problems. Generative Adversarial Networks (GANs) comprise an established category of models which generate realistic data similar to the data of a training data set. In the GAN principle two models are carrying out an adversarial role based on game theory. The generator network tries to fool the discriminator network by sending fake images labelled as true images (training images). The discriminator on the other hand, tries to distinguish between real data (images from the training data set) and fake data (generated images). The training is successful, when the discriminator is no more able to distinguish between the original images and synthetic results producing a classification prediction of 50% for each class.

The generator and the discriminator model are parameterized by deep neural networks. Since we interpret the calorimeter output as a three-dimensional image, we can build neural networks consisting primarily of convolutional layers. Although the generated images are three-dimensional, we designed an architecture which utilizes only 2D convolutional (Conv2D) layers in order to reduce the computational time. The generator architecture is shown in Figure 2 and the discriminator architecture in Figure 3.

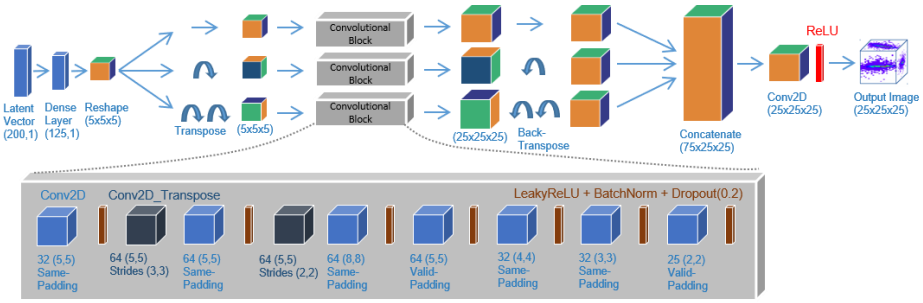

Figure 2. Conv2D generator architecture

The networks consist of three branches corresponding to the three image canonical axes. The generator input latent space comprises *200* random numbers drawn from a uniform distribution between zero and one multiplied by the primary particle energy E_p. The generator output is the three-dimensional image with 25x25x25 pixels. In addition to Conv2D layers, the generator network includes transposed 2D convolutional (Conv2D_transpose) layers to increase the image size, batch normalization (BatchNorm), a rectified linear units activation function (ReLU), linear ReLU activation functions (LeakyReLU) and dropout layers (Dropout). With the help of the three branches, the network is capable to learn the correlations between all three image dimensions.

For the discriminator we employ a model similarly consisting of three branches. The input represents either the real images from the training set or the generated images. The discriminator outputs three values: the first is the typical GAN true/fake probability [7] which is used to calculate a binary cross entropy loss [8]. The second loss (named AUX, for AUXiliary loss) represents the result of a regression task on the initial particle energy E_p, that the discriminator estimates from the images using a dense layer. It is implemented as a Mean Absolute Percentage Error (MAPE) [9]. The third discriminator output comes from a Lambda layer, calculating the sum over the pixels of the input image which, therefore, corresponds to the total energy of the input image. It is entitled ECAL and uses the MAPE loss function likewise.

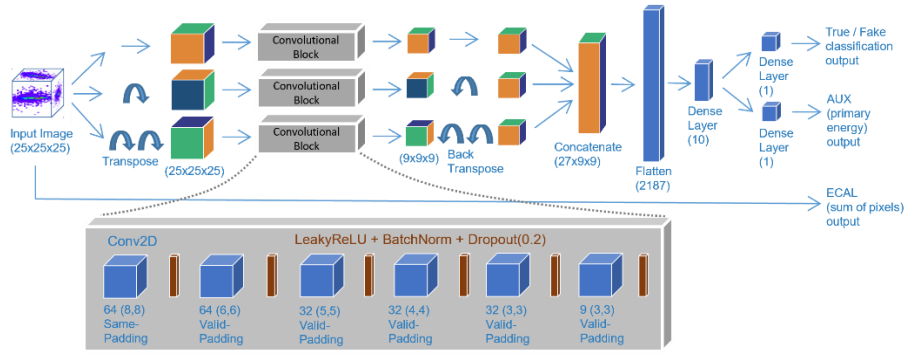

Figure 3. Conv2D discriminator architecture

3. GAN Evaluation

We evaluate the Conv2D GAN model in terms of physics accuracy and computational speed and compare it to a previous architecture taken from Ref. [10] which uses Conv3D layers for the same simulation use case. Ultimately, the new Conv2D model is compared to the Geant4 simulation which is aimed to be replaced. The goal is to speed-up the simulation time while providing the equivalent level of necessary accuracy to evaluate the physics results. The inference is run on a Nvidia Tesla V100 GPU with Python version 3.6.8 and TensorFlow version 2.2.0. We run 20 warm-up batches and evaluate afterwards 100 inference steps including 20 batches each. The inference process of the Conv2D model is optimized with different batch sizes and we measure the speed-up versus Geant4 simulation and the percentage of the GPU utilization. The results are presented in Figure 1. One can see that, for increasing batch sizes, the GPU utilization, and the number of showers per second rises

almost linear until the batch size of *2 048*, where it reaches its peak with *9 347* showers per second. This results in a tremendous *158 000* speed-up compared to the Geant4 simulation which requires 17 seconds to reproduce one single shower image (taken from a previous measurement in Ref. [11]). One can see that at the batch size of *2 048* the GPU is almost completely utilized which results in a drop of showers per second for the measurements with higher batch sizes.

In Table 1 we compare the new Conv2D network to the previous Conv3D architecture. One can see that the Conv2D model provides a much larger speed-up versus Geant4 compared to the Conv3D model, in spite of the fact that the new Conv2D model has a much higher number of parameters and convolutional layers. It should be noted, however, that no batch size optimization was performed for the Conv3D model. However, the GPU utilization of the Conv3D model with a batch size of *128* is already quite high. This is the reason why no significant speed-up of the Conv3D model is expected.

Table 1. The number of parameters and the number of convolutional layers for the Conv2D and Conv3D generator model. The speed-up is given with respect to Geant4 and the last column shows the GPU utilization during inference.

Model	Parameters	Nb. Conv Layers	Speed-up vs Geant4	Utilization
Conv3D	752 000	4	6 200x	78.75%
Conv2D	2 052 000	28	158 000x	98.50%

In order to better quantify the physics agreement of the GAN output with Geant4, we define an accuracy metric based on the mean squared error (MSE). It is calculated by building two-dimensional projections of the particle shower distributions along the *x*-, *y*- and *z*-axis (averaged over *20 000* samples) and measuring the MSE between the corresponding GAN model and Geant4. The Conv2D architecture has an MSE of *0.027* which is lower than the MSE of the previous Conv3D architecture with *0.065* (because this quantity is a measure for the error, the lower the MSE the better the accuracy). The same behavior we can observe in the shower shape plots in Figure 4. (left). The Conv2D model (green) is closer to Geant4 (red) and performs better than the Conv3D model (blue). In particular, the new Conv2D model is able, for the first time, to correctly reproduce the lower energy tails of the shower shape distributions, usually largely overestimated or underestimated by GAN, see Ref. [12].

4. Reduced Precision Research

Modern Deep Learning (DL) dedicated hardware, developed by various vendors to accelerate DL workloads implements different kind of reduced precision strategies. In order to evaluate the effect of reduced precision (int8 in particular) on the inference process of our GAN model, we quantize the neural network parameters from float32 down to the int8 format. We intend to verify whether it is possible to further speed-up the inference and to reduce the memory consumption, while maintaining the physics accuracy. For quantizing model, we use the Intel Low Precision Optimization Tool (LPOT) [13]. LPOT optimizes in an iterative process, based on a predefined accuracy metrics, how many and which weights are quantized. We compare the results with models quantized by the TensorFlow Lite library [14].

We run inference on an Intel 2S Xeon 8280 CPU, "Cascade Lake" architecture, with various numbers of data streams and cores. The best result is achieved with the configuration of 8 streams and 56 cores. We gain a speed-up of *1.8x* from the initial float32 Conv2D model to the int8 Conv2D model (float32 *2 372* showers/second, int8 *4 158* showers/second). On the previously mentioned Intel CPU the speed-up of the quantized int8 model represents *68 000* (different value as in the previous section because it is run on CPU for the research here and on GPU previously) with respect to Geant4. There are multiple reasons why we do not achieve the theoretical expected *4x* speed-up. The first is, that the operations for quantizing of the input and de-quantizing the output takes already 20% of the

computation time. Additionally, the batch normalization layers alone require around 30% of the computation time. In a future LPOT version the batch normalization layer will be combined with the convolutional layer and the activation function which is expected to considerably decrease the simulation time. Due to the quantization, the model memory size is reduced by a factor of *2.26x* from *8.08* MB down to *3.57* MB.

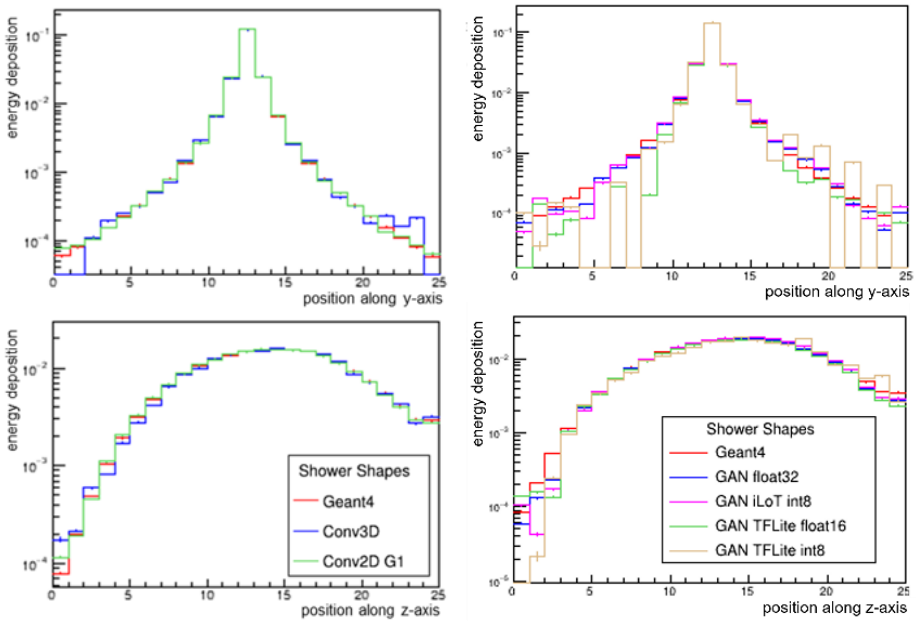

Figure 4. Shower shape plots for measuring physics accuracy. (Left) Comparison of the new Conv2D vs a previous Conv3D architecture. (Right) Quantization of the Conv2D model into lower precision.

Concerning physics accuracy evaluation, we consider the physics metrics introduced in the previous section. The MSE of the initial float32 model is *0.061*, the LPOT int8 is *0.053*, the TFLite float16 is *0.253* and TFLite int8 is *0.340*. One can see, that the quantized LPOT model reaches an even lower MSE and therefore a higher accuracy as the float32 model. This is understood by the fact that the MSE metric was used in the LPOT tool for optimization likewise. Furthermore, the TFLite models perform worse. The reason could be that TFLite quantize the network parameters without any optimization. In Figure 4 (right) the shower distributions are shown for the different quantized models. The LPOT model follows Geant4 very closely, whereas the TFLite models are clearly off for lower energy cells.

5. Conclusion

We introduced a novel Conv2D neural network architecture to successfully solve a 3D image generation task using GANs for the simulation of high granularity calorimeters in HEP experiments. Our GAN model is capable to achieve a tremendous *158 000x* speed-up compared to the Geant4 simulation which we aim to replace. The physics accuracy evaluation demonstrated equally accurate results for the Conv2D GAN model as for Geant4 simulation.

In addition, we investigated the effect of data quantization, from float32 down to the int8 format, using the Intel Low Precision Optimization Tool. We obtained a further *1.8x* speed-up as well as a *2.26x* reduction in model memory size while retaining a good level of physics accuracy.

6. Acknowledgements

This work has been sponsored by the Wolfgang Gentner Programme of the German Federal Ministry of Education and Research.

References

[1] S. Agostinelli, GEANT4--a simulation toolkit, Nucl. Instrum. Meth. A, 2003.

[2] Worldwide LHC Computing Grid [Online]. Available: https://wlcg-public.web.cern.ch/. [Accessed 2021].

[3] F. Rehm, S. Vallecorsa, K. Borras and D. Krücker , "Physics Validation of Novel Convolutional 2D Architectures for Speeding Up High Energy Physics Simulations," 2021.

[4] M. Erdmann, J. Glombitza and T. Quast, "Precise Simulation of Electromagnetic Calorimeter Showers Using a Wasserstein Generative Adversarial Network," in Comput Softw Big Sci 3, 2019.

[5] D. Sipio, Riccardo and Giannelli, "DijetGAN: a Generative-Adversarial Network approach for the simulation of QCD dijet events at the LHC," in Journal of High Energy Physics, 2019.

[6] F. Rehm, S. Vallecorsa, K. Borras and D. Krücker, "Validation of Deep Convolutional Generative Adversarial Networks for High Energy Physics Calorimeter Simulations," in AAAI 2021 - Association for the Advancement of Artificial Intelligence, 2021.

[7] I. Goodfellow, "Generative Adversarial Networks," 2014.

[8] G. E. Nasr, "Cross Entropy Error Function in Neural Networks: Forecasting Gasoline Demand," FLAIRS Conference, 2002.

[9] P. Swamidass, "Encyclopedia of Production and Manufacturing Management," Springer US, pp. 462-462, 2000.

[10] G. Khattak and et al., "Three Dimensional Energy Parametrized Generative Adversarial Networks for Electromagnetic Shower Simulation," 2018.

[11] S. Vallecorsa and F. Carminati, "Distributed Training of Generative Adversarial Networks for Fast Detector Simulation," in High Performance Computing, vol. Springer International Publishing, Springer International Publishing, 2018, pp. 487-503.

[12] G. Khattak, S. Vallecorsa, F. Carminati and M. Khan, "Particle Detector Simulation using Generative Adversarial Networks with Domain Related Constraints," in 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), 2019.

[13] F. Tian and W. Chuanqi, "Intel® Low Precision Optimization Tool LPOT," 2020. [Online]. Available: https://github.com/intel/lp-opt-tool.

[14] "TensorFlow Lite," TensorFlow For Mobile & IoT, [Online]. Available: https://www.tensorflow.org/lite.

[15] F. Rehm, S. Vallecorsa, V. Saletore, H. Pabst , A. Chaibi, V. Codreanu, K. Borras and D. Krücker, "Reduced Precision Strategies for Deep Learning: A High Energy Physics Generative Adversarial Network Use Case," in Proceedings of the 10th International Conference on Pattern Recognition Applications and Methods, 2021.