

Generative models for fast cluster simulations in the TPC for the ALICE experiment

Kamil Deja^{1,*}, Tomasz Trzcíński^{1,**}, and Łukasz Graczykowski²

¹Institute of Computer Science

²Faculty of Physics

Warsaw University of Technology, Pl. Politechniki 1 00-661 Warsaw

Abstract. Simulating the detector response is a key component of every high-energy physics experiment. The methods used currently for this purpose provide high-fidelity results. However, this precision comes at a price of a high computational cost. In this work, we introduce our research aiming at fast generation of the possible responses of detector clusters to particle collisions. We present the results for the real-life example of the Time Projection Chamber in the ALICE experiment at CERN. The essential component of our solution is a generative model that allows to simulate synthetic data points that bear high similarity to the real data. Leveraging recent advancements in machine learning, we propose to use conditional Generative Adversarial Networks. In this work we present a method to simulate data samples possible to record in the detector based on the initial information about particles. We propose and evaluate several models based on convolutional or recursive networks. The main advantage offered by the proposed method is a significant speed-up in the execution time, reaching up to the factor of 10^2 with respect to the currently used simulation tool. Nevertheless, this speed-up comes at a price of a lower simulation quality. In this work we adapt available methods and show their quantitative and qualitative limitations.

1 Introduction

High-energy physics (HEP) experiments, including those conducted at the Large Hadron Collider (LHC) [1], rely heavily on detailed simulations of the detector response in order to accurately compare recorded data with theoretical Monte Carlo models. This simulated data is used to understand the physics behind the real collisions registered in the experiment. The simulations are also used to adjust the experimental data for detector's inefficiencies and various conditions observed in the detector during the data taking procedure.

Traditional approaches to the simulation of the detector response use Monte Carlo methods to generate collisions. To properly simulate events, those methods need to synthesise the propagation of particles through the detector's medium and model every interaction with the experimental apparatus using a transport package, usually GEANT3 [2], GEANT4 [3] or FLUKA [4]. This approach comes with disadvantages. First of all, enlisted packages require specific implementations of the interactions of highly energetic particles with matter that occur inside the detector. In practice, the physics regulating those interactions is complex and

*e-mail: kdeja@mion.elka.pw.edu.pl

**e-mail: t.trzcinski@ii.pw.edu.pl

often difficult to simulate. Secondly, the methodology that relies on Monte Carlo methods is computationally expensive.

To address the above shortcomings of a Monte Carlo-based simulation, we propose a fundamentally different approach to simulations in high-energy physics experiments that relies on generative models. The proof-of-concept solution, that implements this approach is based on the recently proposed conditional Generative Adversarial Networks [5]. The main idea behind using those models for simulations is to learn how to synthesise the situation of the detector after a collision from the data, instead of modelling every interaction of the particles. Such approach enables much faster and easily parallelizable generation.

This work is the extension of the previous article published as a conference proceedings [6]. In [6], we concentrated on the usage of generative models for simulating the possible detector's output. In this work we extend this idea using conditional Generative Adversarial Networks [7]. This approach enables applying additional constrain to generated examples, with initial parameters – in our case describing simulated particles.

As a first step towards implementing our solution in production, we perform a set of experiments that aim at simulating clusters of the Time Projection Chamber (TPC) [8, 9] detector of the ALICE (A Large Ion Collider Experiment) [10] experiment at the LHC. We present quantitative, qualitative and computational cost evaluation of our method for several generative models and their variations, including Deep Convolutional Generative Adversarial Networks [11] and Long Short Term Memory [12] based networks. We also show how to improve the results obtained from those models using combined loss function which leverage from autoencoder and adversarial approaches. To our knowledge, our works represent the first attempt to use machine-learning generative models for TPC clusters simulation in any high-energy physics experiment. The main contribution of this paper is a prototype of a method for event simulation in the TPC detector that relies on conditional generative models. Evaluation results show a promising speedup over currently used Monte Carlo based simulation methods reaching up to 10^2 , however, at the expense of simulation quality.

The remainder of this paper is organised as follows. In the next section, we describe related works. We then overview conditional generative models and present the core idea behind its adaptation. Finally, we describe the dataset and present the results of our evaluation. In the last section, we conclude this work and outline the next steps.

2 Related Work

Currently used methods start from a collision simulation using a Monte Carlo generator, than different transport packages such as GEANT3 [2], GEANT4 [3] or FLUKA [4] are used to synthesise propagation of particles. The desired outputs of transport packages are *clusters*, a set of space-time points calculated from the adjacent sensitive elements of the detector that were presumably generated by the same particle crossing these elements.

Machine learning generative models, which we intend to use in the context of high-energy physics, have already proven their potential in numerous research areas mostly related with image processing. However, there are several attempts to use Generative Adversarial Networks and especially conditional ones, for simulating parts of the physics processes, like the very first one in the field: calorimeters response simulation [13]. In that work, the authors generate particle showers in electromagnetic calorimeters. As a generalisation of this approach, in [14] authors propose a holistic approach of generating particle showers in numerous detectors. Although our work draws inspiration from [13] and [14], we tackle a different problem related to the particle trajectories defined as consecutive particle clusters, while both of the listed methods focuses on generating particle showers which are three dimensional images.

3 Generative models

3.1 Generative Adversarial Networks

The main concept behind Generative Adversarial Network [5] is to train two neural networks to play a *min-max game* between each other.

The first one – a *generator* – is trained to generate new data points whose distribution resembles the distribution of real data. The second one – trained as a *discriminator* – aims at predicting whether a given example comes from a real or synthetic distribution. During the training process, both networks are updated iteratively, one at a time, which leads to a simultaneous optimization of loss functions of both generator and discriminator.

The *min-max* loss function introduced in [5], encompassing both networks can be written in the following manner presented in the Eq. 1:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (1)$$

where $p_{data}(x)$ is the distribution of real data, x is a sample from p_{data} , $p(z)$ is a distribution of a noise vector z input into a generator and G and D represent a discriminator.

3.2 Conditional Generative Adversarial Networks

the basic Generative Adversarial Network model does not allow to incorporate any additional information about generated examples. However, in some use cases there is an explicit indicator which conditions the distribution of desired output. Therefore the Conditional Generative Adversarial Networks [7] emerged to overcome these limitations.

In conditional approach authors suggest to include additional information to the standard GAN in two steps. Firstly, the noise applied as a generator input is generated with the respect of initial parameters. Secondly, the same values are applied to the discriminator so it can distinguish not only if the generated sample is similar to the real data, but also if it fits the provided parameters. In our work, we benefit from this solution, using additional, initial information about simulated particles.

4 Dataset

To evaluate the performance of generative models, we use a dataset of 3D trajectories of particles after collision generated with a Monte Carlo simulation method of PYTHIA 6.4 [15] Perugia-0 [16] model. The dataset therefore contains a sample of proton–proton collisions at the centre of mass energy of $\sqrt{s} = 7$ TeV.

After the simulated collision, the generated particles were transported, using GEANT3 [2] package, through the detector’s medium. The trajectories correspond to the real traces observed in the TPC detector of the ALICE Experiment at LHC, CERN and the experimental conditions of 2010 data-taking period.

Since the size of the dataset generated this way results in petabytes of data, we randomly sample it to contain around 300 events corresponding to approximately 1.000.000 data samples. Though, from further experiments we excluded all of the particles which were not produced exactly during the collision. We also removed examples with less than 50 clusters, because of their little importance for physical analysis. Finally after this preprocessing we prepared dataset of 48523 data samples.

Each data sample representing particle clusters is stored together with the initial particle’s momenta (p_x, p_y, p_z) its charge, and mass, as the initial values generated by the monte carlo simulation. This input is associated with a series of 3D coordinates (x, y, z) corresponding to the trajectory of a given particle after the collision. The minimum and maximum values of the coordinates depend on the detector size, which is a cylinder-shaped structure of approximately $5m \times 5m \times 5m$. Since the collision does not necessarily occurs in the central point of

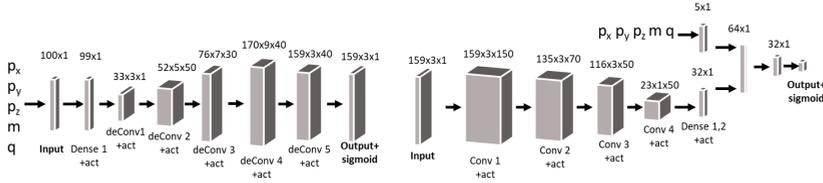


Figure 1. Architecture for the conditional DCGAN model. Each block represent a network’s layer with its size given above. Network is trained on two individual inputs – generated noise and particle parameters

the detector, the minimal distance from the $[0, 0, 0]$ point, where the trajectories are collected, is 848 mm. The precision of the recorded coordinates is limited to the resolution of the TPC read-out pads, which is between 0.8 to 1.2 mm, depending on the size of the pad [8]. For normalisation purposes, the input data coordinates are scaled to fit the $[0, 1]$ interval in each dimension. We also apply other normalisation procedures, such as zero-padding the samples for particle trajectories consisting less than 159 points (maximum number which can be registered by readout pads). Although additional characteristics of the particles can be observed, *e.g.* its energy and speed, for the purpose of the evaluation presented in this work we restrict our data samples to 3D coordinate values. Therefore we use the data with final resolution of 159×3 pixels. The initial noise inputted to the generator consisted of **100** random values.

After the collision occurs, up to few hundreds particle trajectories can be observed inside the detector. Although we could attempt to simulate all of those trajectories at once using generative models, we postulate to generate separate trajectories for individual particles first and then merge them together to achieve the final goal. The highest reported resolution of the output generated by GANs, is 1024×1024 [17], and it is relatively low, when compared to the one observed in the TPC detector – $5000 \times 5000 \times 5000$ possible clusters locations. To circumvent this limitation, we transform our simulation problem into a so-called transactional form and simulate individual points with their (x, y, z) coordinates, which we can then link together to form a full trajectory.

5 Architectures

The general framework of Generative Adversarial Networks is rather flexible in terms of the type and number of layers used as a generator and discriminator. Hence, we propose several architectures that fit to the context of simulating detector’s response in a high-energy physics experiment, with respect to the dataset described above.

- **Conditional Deep Convolutional Generative Adversarial Network (condDCGAN):** multi layered network, with two dimensional convolutional/de-convolutional layers, that employs initial particles information, as shown in Fig. 1.
- **Conditional LSTM based Generative Adversarial Network (condLSTMGAN):** multi layered recurrent network which uses LSTM units to process recursive data.

5.1 Conditional Generative Adversarial Network with additional loss

To enhance the quality of the samples generated with conditional GAN, as well as to prevent the mode collapse we propose the new training procedure, which derives from both Generative Adversarial Networks and Variational Autoencoders. The idea is to train the generator to generate new samples based only on the parameters selected from the training data. It allows to compare generated samples with the original ones in the same manner as in autoencoder. However, to ensure the non-deterministic behaviour of the model, we propose to combine

the distance measured from original sample with the standard loss produced by the GAN's discriminator. The exact generator's loss function for such an approach is described in Eq. 2.

$$\mathcal{L}_G(m, X) = \mathbb{E}_{z \sim p_z(z|m)} [\alpha \log(1 - D(G(z))) + \beta \frac{1}{n} \sum_{i=1}^n (X_i - G(\hat{z}_i))^2] \quad (2)$$

where: m is the initial parameters (particle momenta), X an original value corresponding to m , $p(z|m)$ is a distribution of a noise vector under initial parameters m , z input into a generator, and G and D represent generator and discriminator. In the second part of the loss function n is the number of produced clusters. Additional parameters α and β are used to weight the share of individual losses.

We conduct tests on this training procedure using three architectures. In order to distinguish them during the evaluation process, we refer to them using additional pluses at the end of architecture's name (condLSTM GAN+ and condDCGAN+). In addition, to speed up the training, and to make sure that the discriminator learns to distinguish whether the given output is in line with initial parameters, we conduct additional discriminator training every 2000 batches.

5.2 Implementation details

To train all networks, we use dropout [18], Leaky ReLu activation [19], and batch normalization [20]. As our loss function, we take a binary cross-entropy loss and non-saturating loss. We optimize our loss function with ADAM, a stochastic optimization method [21]. We initialize network weights with the glorot algorithm [22]. The neural networks are implemented in Python using Keras library [23] with TensorFlow backend [24]. Our implementations are based on those enlisted in the large scale study [25].

6 Results

In this section, we evaluate methods based on generative models in terms of both: the quality of the generated samples and their computational cost. As the baseline model, we use the Monte Carlo-based methods currently used in the ALICE Experiment to generate the full TPC response simulation of the generated particle showers, propagated through the detectors medium by the GEANT3 package. We use the most recent implementation provided as part of the O2 software [9, 26].

To evaluate the quality of the generated trajectories, we refer to the physical property of the particle tracks observed after the collision. When particles move through the detector medium, they form a track whose shape can be defined as a helix. We leverage this phenomenon and calculate for each method an evaluation metric that measures the distance of generated trajectory from a theoretically ideal shape of helix, approximated with an arc. We report this metric as a mean squared error (MSE) between the ideal helix shape and the simulated points to average it across all simulated particles. For proper evaluation of conditional models' quality we randomly split our initial dataset to two separate fragments: training set with 40 000 and test set with remaining 14 872 data samples.

We also evaluate the computation cost of generating the results for various methods. To that end, we execute the code 10 times on a standalone machine with an Intel Core i7-6850K (3.60GHz) CPU (using single core, no GPU acceleration) and record the average execution time. We then normalise it with respect to the baseline method of simulation, namely GEANT3, which is currently used to simulate the detector's response.

Table 1 presents the results of simulation quality evaluation for different models based on conditional GAN approach. Each model was trained for 200 000 batches of data randomly sampled from trainset which is equal to around 200 epochs on the whole training data. Training of the best models using one Titan XP GPU took several hours.

Table 1. Quality of conditional generative models, comparing to the GEANT3 simulation.

Method	Mean MSE (mm)	Median MSE (mm)	speed-up
GEANT3 (<i>current simulation</i>)	1.20	1.12	1
Random (<i>estimated</i>)	2500	2500	N/A
condLSTM GAN	2093.69	2070.32	10^2
condLSTM GAN+	221.78	190.17	
condDCGAN	795.08	738.71	25
condDCGAN+	136.84	82.72	

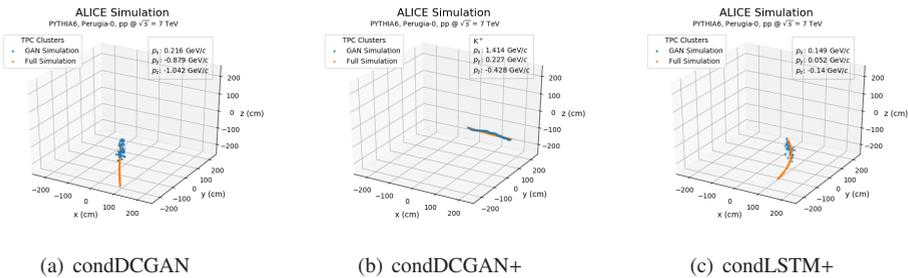


Figure 2. Exemplar results generated by different models (a) conditional DCGAN without additional loss, (b) conditional DCGAN and (c) conditional LSTM GAN, with additional loss

We compare the results obtained with standard conditional GAN approach with those generated with additional loss function (additional + sign by the name). There is no difference in architectures of enhanced and standard models therefore the observed speed-up for generating new samples is equal. As denoted in table 1 there are visible problems with **condLSTM GAN** training. Without additional loss, the generator could not learn well the distribution of real data. Hence it produced almost random clusters. As listed in table 1, the mean MSE is only slightly lower than for the randomly generated samples. However, application of additional loss in condLSTM GAN+ model resulted in its better convergence and made it possible to properly train the generator. The MSE error was therefore reduced to around 200mm. As presented in Fig. 2(c) the generator well resolved the direction in which the particle flew however, the simulated track is fuzzy what explains the high MSE value of the model. Nevertheless the condLSTM GAN model is rather shallow, hence the observed speed-up is significant and reaches 2 orders of magnitude.

When analysing the results obtained with convolutional conditional models, the standard **condDCGAN** converges better than the LSTM based one. In evaluation conducted on the testset we observed the median MSE value of 738.71 millimetres. As visible in Fig. 2(a) this time there are also evident problems with track’s cohesion.

To improve the accuracy of results produced with condDCGAN model we applied additional cost to the generator, which penalised it for creating clusters in some distance from the original ones. Application of this idea in condDCGAN+ model reduced the median MSE value to around 82.72 millimetres. Considering that the detector’s resolution reaches around 1 mm, the 8.3cm discrepancy is rather big. However, as presented in Fig. 2(b), for certain type of data the two tracks – generated and original ones are almost identical, even though, the generated clusters are not equal. We debate over this phenomenon later however, it may suggest that the solution proposed in this work can be applied at least for some part of pro-

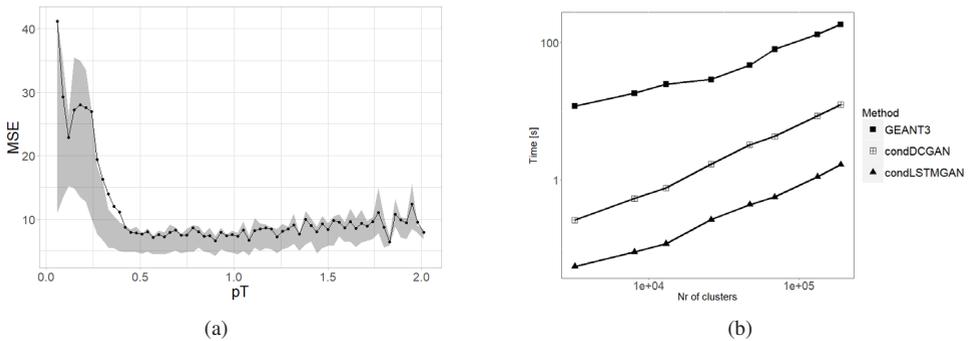


Figure 3. Mean MSE per particle’s perpendicular momentum p_T . Grey area represents the first and third quartile of MSE. (left) and Performance evaluation for conditional GAN models (right)

cessed data. The main reason for such an application is that condDCGAN is able to generate results around 25 times faster than standard GEANT method.

Analysing the results obtained with best performing **condDCGAN+** model, we discovered that the greater the particle initial momentum is, the higher is the accuracy of cluster’s position generated with generative model. It is especially visible when concerning momenta perpendicular to the detectors axis dubbed p_T . The results of the analysis are presented in Fig. 3(a). It is known, that the greater the momentum, the straighter the particle track, hence probably generative models evaluated in this work tend to produce straighter tracks with not sufficient curvatures for particles with lower momenta.

Despite the precision of our methods, to emphasise their true potential we present the execution times for different methods, while increasing the number of simulated clusters, i.e. model outputs. Fig. 3(b) shows the results of this experiment. Although the computational cost of all the methods increases linearly with the number of simulated detector responses, the improvement achieved by the generative models is massive. It is worth mentioning that the presented results refers to the execution on a single-core CPU, while additional hardware-based speed-up of another order of magnitude was observed when using the GPU-based implementation for the neural network methods. Although Monte Carlo simulations can also benefit from such an acceleration, its iterative character result in much lower speed-up [27].

7 Summary

In this work, we demonstrated the potential of machine learning generative methods namely conditional Generative Adversarial Networks for cluster simulation in the high-energy physics experiments. We proved the possible application of those methods using the example dataset generated for the TPC detector in the ALICE Experiment at LHC. We implemented and evaluated several architectures, and compared their results in terms of quality and computational cost. We also introduced the new training procedure which proved to provide more precise results in faster way than the standard approaches. The quality of the best performing method is not yet equal to the quality of currently used simulation methods. However, the analysis of results revealed that proposed solutions can well reproduce the whole events while maintaining the laws of physics. The most important advantage is that described methods are much faster. The observed computational speed-up is unprecedented and reaches up to 10^2 when compared to the currently employed GEANT 3 simulation technique.

Understanding the limitations of proposed methods, we plan to extend our works to reduce the inaccuracy of generated examples. With increased accuracy we also plan to use the presented models as a main component of the semi-real time anomaly detection tool.

8 Acknowledgements

The authors acknowledge the support from the Polish National Science Centre grant no. UMO-2016/21/D/ST6/01946. The GPUs used in this work were funded by the grant of the Dean of the Faculty of Electronics and Information Technology at Warsaw University of Technology (project II/2017/GD/1).

References

- [1] L. Evans, P. Bryant, *JINST* **3**, S08001 (2008)
- [2] R. Brun, F. Bruyant, F. Carminati, S. Giani, M. Maire, A. McPherson, G. Patrick, L. Urban, CERN-W5013, CERN-W-5013, W5013, W-5013 (1994)
- [3] S. Agostinelli et al. (GEANT4), *Nucl. Instrum. Meth.* **A506**, 250 (2003)
- [4] A. Ferrari, P.R. Sala, A. Fasso, J. Ranft, CERN-2005-010, SLAC-R-773, INFN-TC-05-11 (2005)
- [5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, in *NIPS* (2014)
- [6] K. Deja, T. Trzcinski, L. Graczykowski, for the ALICE Collaboration, *Generative Models for Fast Cluster Simulations in the TPC for the ALICE Experiment*, in *3rd Conference on Information Technology, Systems Research and Computational Physics* (2018)
- [7] M. Mirza, S. Osindero, arXiv preprint arXiv:1411.1784 (2014)
- [8] G. Dellacasa et al. (ALICE), CERN-OPEN-2000-183, CERN-LHCC-2000-001 (2000)
- [9] B. Abelev et al. (ALICE), *J. Phys.* **G41**, 087001 (2014)
- [10] K. Aamodt et al. (ALICE), *JINST* **3**, S08002 (2008)
- [11] A. Radford, L. Metz, S. Chintala, CoRR **abs/1511.06434** (2015)
- [12] S. Hochreiter, J. Schmidhuber, *Neural computation* **9**, 1735 (1997)
- [13] M. Paganini, L. de Oliveira, B. Nachman, CoRR **abs/1705.02355** (2017)
- [14] G.R. Khattak, S. Vallecorsa, F. Carminati, *Three Dimensional Energy Parametrized Generative Adversarial Networks for Electromagnetic Shower Simulation* (2018)
- [15] T. Sjostrand, S. Mrenna, P.Z. Skands, *JHEP* **05**, 026 (2006), hep-ph/0603175
- [16] P.Z. Skands, *Phys. Rev.* **D82**, 074018 (2010), 1005.3457
- [17] T. Karras, T. Aila, S. Laine, J. Lehtinen, CoRR **abs/1710.10196** (2017)
- [18] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, *JMLR* **15**, 1929 (2014)
- [19] X. Glorot, A. Bordes, Y. Bengio, *Deep sparse rectifier neural networks*, in *International Conference on Artificial Intelligence and Statistics* (2011), pp. 315–323
- [20] S. Ioffe, C. Szegedy, *Batch normalization: Accelerating deep network training by reducing internal covariate shift*, in *ICML* (2015)
- [21] D.P. Kingma, J. Ba, CoRR **abs/1412.6980** (2014)
- [22] X. Glorot, Y. Bengio, *Understanding the difficulty of training deep feedforward neural networks*, in *International Conference on Artificial Intelligence and Statistics* (2010), pp. 249–256
- [23] F. Chollet et al., *Keras*, <https://github.com/fchollet/keras> (2015)
- [24] M. Abadi et al., *TensorFlow: A system for large-scale machine learning*, in *(OSDI 16)* (2016), pp. 265–283
- [25] M. Lucic, K. Kurach, M. Michalski, S. Gelly, O. Bousquet, CoRR **abs/1711.10337** (2017)
- [26] A.A.D.P. Suaide, C.A.G. Prado, T. Alt, L. Aphecetche, N. Agrawal, A. Avasthi, M. Bach et al., *O2: A novel combined online and offline computing system for the ALICE Experiment after 2018*, in *Journal of Physics: Conference Series* (2014), Vol. 513, p. 012037
- [27] O. Seiskari, J. Kommeri, T. Niemi, arXiv preprint arXiv:1209.5235 (2012)