Generative Models for Fast Calorimeter Simulation: the LHCb case

Viktoria Chekalina[1,2], *Elena* Orlova[3], *Fedor* Ratnikov[1,2,*], *Dmitry* Ulyanov[3], *Andrey* Ustyuzhanin[1,2], and *Egor* Zakharov[3]

[1]NRU Higher School of Economics, Moscow, Russia
[2]Yandex School of Data Analysis, Moscow, Russia
[3]Skolkovo Institute of Science and Technology, Moscow, Russia

Abstract. Simulation is one of the key components in high energy physics. Historically it relies on the Monte Carlo methods which require a tremendous amount of computation resources. These methods may have difficulties with the expected High Luminosity Large Hadron Collider (HL-LHC) needs, so the experiments are in urgent need of new fast simulation techniques. We introduce a new Deep Learning framework based on Generative Adversarial Networks which can be faster than traditional simulation methods by 5 orders of magnitude with reasonable simulation accuracy. This approach will allow physicists to produce a sufficient amount of simulated data needed by the next HL-LHC experiments using limited computing resources.

1 Introduction

Simulation plays an important role in particle and nuclear physics. It is widely used in detector design and in comparisons between experimental data and theoretical models. Traditionally, simulation relies on *Monte Carlo methods* and requires significant computational resources. In particular, such methods do not scale to meet the growing demands resulting from large quantities of data expected during High Luminosity Large Hadron Collider (HL-LHC) runs. The detailed simulation of particle collisions and interactions as captured by detectors at the LHC using a well-known simulation software `Geant4` annually requires billions of CPU hours constituting more than half of the LHC experiments' computing resources [1, 2]. More specifically, the detailed simulation of particle showers in calorimeters is the most computationally demanding step.

A line of simulation methods that exploit the idea of reusing previously calculated or measured physical quantities have been developed to reduce the computation time [3, 4]. These approaches suffer from being specific to an individual experiment and, despite being faster than the full simulation, they are not fast enough or lack accuracy. Thus, the particle physics community is in need of new faster simulation methods to model experiments.

One of the possible approaches to simulate the calorimeter response is using *deep learning* techniques. In particular, a recent work [5], provided evidence that *Generative Adversarial Networks* can be used to efficiently simulate particle showers. While over $100,000\times$ speed-up over `Geant4` is achieved, the setup was quite simple as the input particles were

[*]e-mail: fedor.ratnikov@cern.ch

parametrized by energy only. However, even in this simplified approach, there are significant differences in distributions between generated and original parameters.

In this work we build a model upon Wasserstein Generative Adversarial Networks and show its superior performance over approach [5]. We also evaluate our model in a more complex scenario, when a particle is described by 5 parameters: 3d momentum (p_x, p_y, p_z) and 2d coordinate (x, y). Our method for high-fidelity fast simulation of particle showers in the specific LHCb calorimeter aims to replace the existing Monte Carlo based methods and achieve a significant speed-up factor.

2 Related work: GANs basics and GANs in HEP

Generative models are of great interest in deep learning. With these models, one can approximate a very complex distribution defined as a set of samples. For example, such models can be utilized to generate a face image of a non-existing person or to continue a video sequence given several initial frames. In this section, we give a brief overview of the most popular generative model in computer vision — Generative Adversarial Networks (GANs), its strong and weak sides and different modifications to alleviate its weaknesses. Then, we review and analyse current approaches for applying GANs to the simulation of calorimeters in High energy physics.

2.1 Background: from GAN to conditional WGAN

Generative Adversarial Networks (GANs) were originally presented by I. Goodfellow *et al.* in 2014 [6] and quickly became a state-of-the-art technique in areas such as image generation [7], with a huge number of extensions [8–10].

In the GAN framework, the aim is to learn a mapping G, usually called *generator*, to warp an easy-to-draw distribution $p(\mathbf{z})$ (e.g. $p(\mathbf{z}) = \mathcal{N}(0, I)$) into a target distribution $p_{\text{data}}(\mathbf{x})$ to facilitate sampling from $p_{\text{data}}(\mathbf{x})$. When G is learned, $G \equiv G^*$, sampling from the target distribution $p_{\text{data}}(\mathbf{x})$ is done by first drawing a sample from the distribution $p(\mathbf{z})$ and then feeding the sample into the generator: $G^*(\mathbf{z}) \sim p_{\text{data}}$, where $\mathbf{z} \sim p(\mathbf{z})$. For such sampling procedure, the time needed to draw a sample from $p_{\text{data}}(\mathbf{x})$ is approximately equal to the time needed to evaluate the function G in a point.

The generator is learned by using a feedback from an external classifier (usually called *discriminator*), which tries to find discrepancy between the target distribution $p_{\text{data}}(\mathbf{x})$ and fake distribution $p_G(\mathbf{x})$ defined by samples from the generator $G(\mathbf{z}) \sim p_G(\mathbf{x})$, $\mathbf{z} \sim p(\mathbf{z})$.

More formally, generator G and discriminator D play the following zero sum game:

$$\min_G \max_D \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{x}\sim p_G(\mathbf{x})}[\log(1 - D(\mathbf{x}))]\,, \tag{1}$$

where $D(G(\mathbf{z}))$ is the output of the discriminator specifying the probability of its input to come from the target distribution p_{data}.

In practice, the mappings G and D are parametrized by deep neural networks and the objective Eq. (1) is optimized using alternating gradient descent. For a fixed generator, the discriminator minimizes binary cross-entropy in a binary classification problem (samples from p_{data} versus samples from p_G). For the fixed discriminator, the generator is updated to make its samples to be misclassified by the discriminator, thus moving the fake distribution closer to the target distribution.

For a fixed generator, it is possible to show that the optimal value for the inner optimization can be written analytically:

$$\max_D \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{x}\sim p_G(\mathbf{x})}[\log(1 - D(\mathbf{x}))] = \text{JS}(p_{\text{data}} \parallel p_G)\,, \tag{2}$$

where JS is the Jensen-Shannon divergence. In fact, for the fixed generator (hence fixed fake distribution), the discriminator computes the divergence between the target distribution p_{data} and the fake distribution p_G. When the divergence is computed, the generator aims to update the fake distribution to make this divergence lower: $\min_G \text{JS}(p_{\text{data}} \parallel p_G)$. While the Jensen-Shannon divergence naturally arises from the original game Eq. (1), any divergence or distance $\mathcal{D}$ can be used instead: $\min_G \mathcal{D}(p_{\text{data}} \parallel p_G)$. A recent work [11] proposed to use the *Wasserstein distance* instead of the Jensen-Shannon divergence proving its better behavior:

$$W(p_{\text{data}} \parallel p_G) = \max_{f \in \mathcal{F}} \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})}[f(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \sim p_G}[f(\mathbf{x})] \tag{3}$$

where $\mathcal{F}$ is a set of 1-Lipshitz functions. Using the Wasserstein distance instead of the Jensen-Shannon divergence in the GAN objective leads to the Wasserstein GAN (WGAN) objective:

$$\min_G \max_{f \in \mathcal{F}} \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})}[f(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \sim p_G(\mathbf{x})}[f(\mathbf{x})]\,. \tag{4}$$

It is highly non-trivial to search over the set of 1-Lipshitz functions and several ways have been proposed in order to force this constraint [11, 12]. In Ref. [12], it is proved that the set of optimal functions for Eq. (4) contains such function, that the norm of it's gradient in any point equals one. In practice, this result motivates an additional loss added to the objective Eq. (4) with a weight λ, while the hard constraint on the function f to belong to the set $\mathcal{F}$ is removed and f is searched over all possible functions:

$$\min_G \max_f \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} f(\mathbf{x}) - \mathbb{E}_{\mathbf{x} \sim p_G(\mathbf{x})} f(\mathbf{x}) + \lambda \mathbb{E}_{\mathbf{x} \sim p_G} (\|\nabla_{\tilde{\mathbf{x}}} D(\tilde{\mathbf{x}})\|_2 - 1)^2. \tag{5}$$

WGAN can be easily adapted to model a conditional distribution $p_{\text{data}}(\mathbf{x}|\mathbf{y})$. The generator is modified to take the condition along with the sample $\mathbf{z}$ so the fake distribution is now defined as $G(\mathbf{z}, \mathbf{y}) \sim p_G(\mathbf{x}|\mathbf{y})$, $\mathbf{z} \sim p(\mathbf{z})$ and the game is

$$\min_G \max_f \mathbb{E}_{\mathbf{y} \sim p(\mathbf{y})} \Big[\mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x}|\mathbf{y})} f(\mathbf{x}) - \mathbb{E}_{\mathbf{x} \sim p_G(\mathbf{x}|\mathbf{y})} f(\mathbf{x}) + \lambda \mathbb{E}_{\mathbf{x} \sim p_G(\mathbf{x}|\mathbf{y})} (\|\nabla_{\tilde{\mathbf{x}}} D(\tilde{\mathbf{x}})\|_2 - 1)^2 \Big]. \tag{6}$$

2.2 GANs in high energy physics

A systematic study on the application of deep learning to the simulation of calorimeters for particle physics has been carried out by Paganini et al. in 2017 [5] and has resulted in the CaloGAN package. The authors aim to speed up particle simulation in a 3-layer heterogeneous calorimeter using GANs framework and achieve $\sim \times 10^5$ speedup. They used an existing state-of-the-art but slow simulation engine `Geant4` to create a training dataset. They simulated positrons, photons and charged pions with various energies sampled from a flat distribution between 1 GeV and 100 GeV. All incident particles in this study have an initial momentum perpendicular to the face of the calorimeter. The shower in the first layer is represented as a 3×96 pixel image, the middle layer as a 12×12 pixel image, and the last layer as a 12×6 pixel image.

Their design of the generator network is based on a DCGAN structure [7] with some convolutional layers replaced by locally-connected layers [13]. The idea of locally connected layers is based on the fact that every pixel position gets its own filter while an ordinary convolutional layer is applied over the whole image, independently of location. An extension of this method to particle physics simulation has been described in the previous work of the authors, where the resulting type of neural network was called LAGAN [14]. A special section in the paper is devoted to the evaluation of the quality of the CaloGAN produced images, where the sparsity level, energy per layer or total energy, are used as measures of the performance of the model.

The obtained results demonstrate a prospect of application of GANs for the particle showers generation and its replacement of the Monte Carlo methods with the proposed approach. The CaloGAN approach yields sizeable simulation-time speedups compared to `Geant4` .

3 Dataset

In this work, we focused on electrons interactions inside an electromagnetic calorimeter inspired by the LHCb detector at the CERN LHC [15]. The calorimeter in this study uses "shashlik" technology of alternating scintillating tiles and lead plates. The prototype consists of 5 × 5 blocks of size 12 cm × 12 cm, the cell granularity corresponds to each block being 6 × 6 of size 2 cm × 2 cm. There are 66 total layers in ECAL, 2 mm lead absorber and 4 mm scintillator each. In fact, the shower appears in 3d, but all energies deposited in all scintillator layers of one cell are summed up. This procedure reproduced the actual shower energy collection in the calorimeter. Thus, the calorimeter response can be represented as 30 × 30 images Y with the corresponding parameters $(p_x,\ p_y,\ p_z,\ x,\ y)$ of the original particle. An example of such an image is presented in the top row of Fig. 3.

The training data set is created as follows. The calorimeter prototype structure described above is described in `Geant4` as a mixture of subsequent sensitive and insensitive volumes. Particles are generated using a particle gun. Particle energies are distributed dropping as $1/E$ in the energy range between 1 and 100 GeV. Particle positions are generated uniformly in the square 1×1 cm in the centre of the calorimeter face. Finally, particle angles are distributed normally with widths of 20 degrees in XZ plane and 10 degrees in YZ plane. Then `Geant4` is used to simulate particle interaction with the calorimeter using the full set of corresponding physics processes. Information about every event, therefore, includes the original particle parameters accompanied by 30 × 30 matrix of energies deposited in scintillators for every cell tower Y. Electrons are used as test particles. Produced training dataset contains 50 000 events, and another 10 000 events are used as a test data sample.

4 Our GAN model

Our idea is to treat simulations as a black-box and replace the traditional Monte Carlo simulation with a method based on Generative Adversarial Networks. As WGANs with gradient penalty are considered to be the state-of-the-art technique for image producing, we implement a tool based on this approach. For it to be useful in realistic physics applications, such a system needs to be able to accept requests for the generation of showers originating from incoming particle parameters such as 3d momentum and 2d coordinate. We introduce an auxiliary task of reconstructing these parameters p_x, p_y, p_z and x, y from a shower image.

4.1 Model architecture

We need to generate a specific calorimeter response for a particle with some parameters. It means that the model is required to be conditional. Firstly, we describe a generator and discriminator architecture. The generator maps from an input (a 512 × 1 vector sampled from a Gaussian distribution and the particle parameters) to a 30 × 30 image $\hat{\mathbf{y}}$ using deconvolutional layers (in fact, it is an upsampling procedure and convolutions) which are arranged as follows. We concatenate the noise vector and the parameters $(p_x,\ p_y,\ p_z,\ x,\ y)$, after which we add a fully connected layer with reshaping and obtain a 256 × 4 × 4 output. After a sequence of 2d deconvolutions, we get outputs of size 128 × 8 × 8, 64 × 15 × 16 and 32 × 32

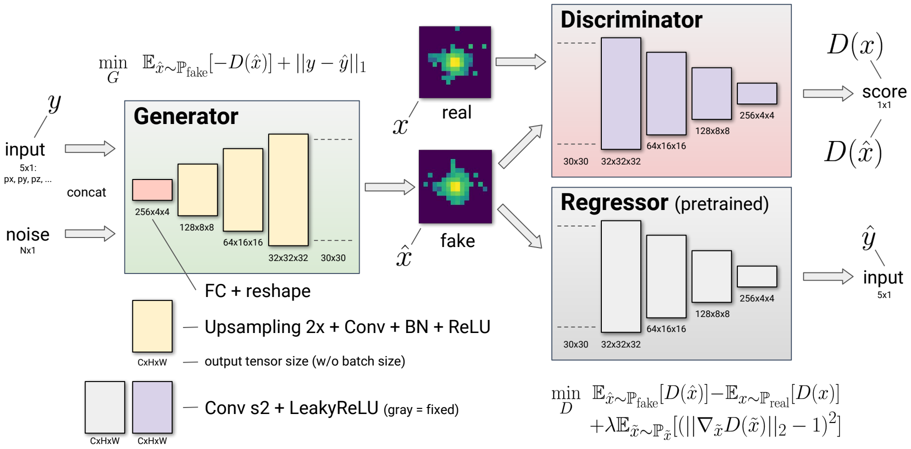

$\min_G \; \mathbb{E}_{\hat{x}\sim\mathbb{P}_{\text{fake}}}[-D(\hat{x})] + ||y-\hat{y}||_1$
y
input
5x1:
px, py, pz, ...
concat
noise
Nx1
Generator
256x4x4
128x8x8
64x16x16
32x32x32
30x30
x
real
$\hat{x}$
fake
Discriminator
30x30
32x32x32
64x16x16
128x8x8
256x4x4
$D(x)$
score
1x1
$D(\hat{x})$
Regressor (pretrained)
30x30
32x32x32
64x16x16
128x8x8
256x4x4
$\hat{y}$
input
5x1
FC + reshape
Upsampling 2x + Conv + BN + ReLU
CxHxW
output tensor size (w/o batch size)
CxHxW
CxHxW
Conv s2 + LeakyReLU (gray = fixed)
$\min_D \; \mathbb{E}_{\hat{x}\sim\mathbb{P}_{\text{fake}}}[D(\hat{x})] - \mathbb{E}_{x\sim\mathbb{P}_{\text{real}}}[D(x)]$
$+\lambda\mathbb{E}_{\tilde{x}\sim\mathbb{P}_{\tilde{x}}}[(||\nabla_{\tilde{x}}D(\tilde{x})||_2 - 1)^2]$

Figure 1: Model architecture. Pre-trained regressor for the particle parameters prediction makes our model conditional. Thanks to building up the information from the pre-trained regressor into the discriminator gradient we learn G to produce a specific calorimeter response.

× 32 with ReLu activation functions. After this procedure, we crop the last output to obtain the image of the desired size 30 × 30.

As for the discriminator, it takes a batch of images as input (all images in the batch are real or generated by G) and returns the score $D(\mathbf{y})$ or $D(\hat{\mathbf{y}})$ as it is described in [11]. The discriminator architecture is simply the reversed generator architecture (i.e. sizes of layers go in the opposite order). It implies that we have a 30 × 30 matrix as input, from which we obtain output layers of size 32 × 32 × 32, 64 × 15 × 16, 128 × 8 × 8, followed by reshaping, which leads to 256 × 4 × 4, and by applying LeakyRelu activation function we get the final score. The model scheme is presented in Fig. 1.

How to train WGAN with gradient penalty in a conditional manner is described in the following section.

4.2 Training strategy

Due to the nature of WGAN loss, conditioning on the continuous value is a non-trivial task. To overcome this issue we suggest embedding a pre-trained regressor in our model. We train a neural network to predict the particle parameters by the calorimeter response. As for architecture, it has the same one as the discriminator but with a perceptual loss described in [16], because it was seen to work better compared to standard MSE. By building up the information from the pre-trained regressor into the discriminator gradient, we obtain the conditional model because we train the generator and the discriminator together. As a result, the discriminator makes the generator produce a specific calorimeter response.

Matrices from our dataset are pretty sparse because almost all information is located in central cells (see Fig. 2). To make the optimization process easier we apply a box-cox transformation. This mapping helps to smooth the data that makes the optimization process more stable. Results obtained with the described model are presented in the following section.

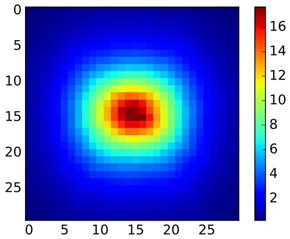
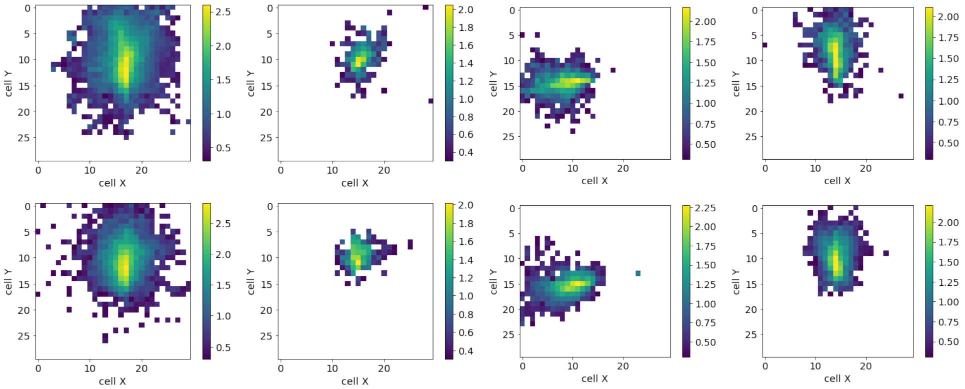


Figure 2: energy deposition in different cells of used 30×30 setup for `Geant4` simulated events averaged over all events in the used dataset.



(a)
$E_0 = 63.7$ GeV

(b)
$E_0 = 6.5$ GeV

(c)
$E_0 = 15.6$ GeV

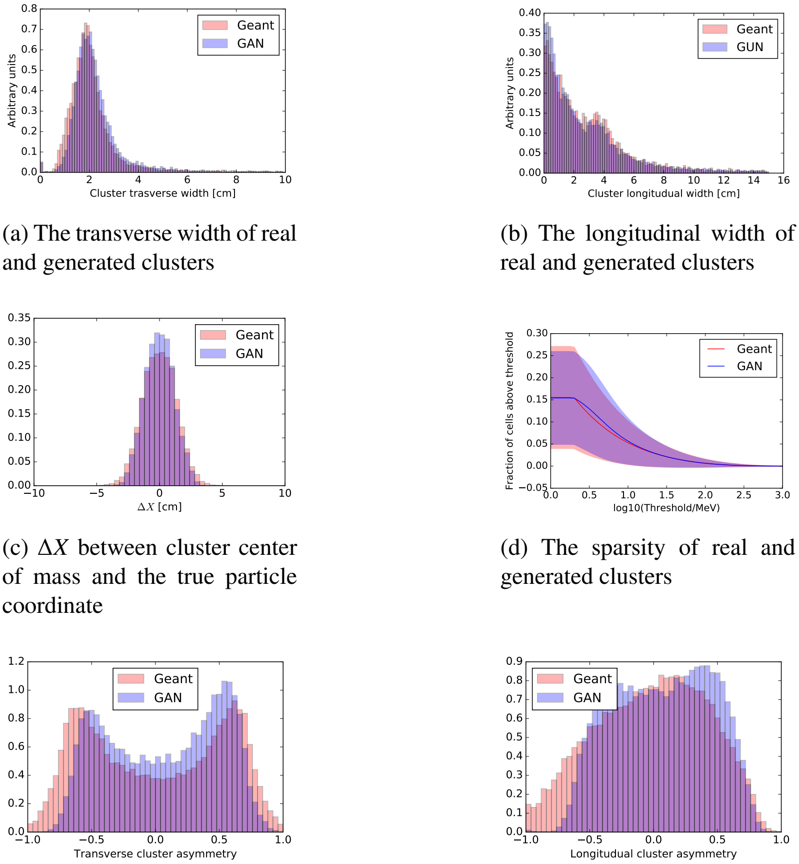
(d)
$E_0 = 15.9$ GeV

Figure 3: Showers generated with `Geant4` (first row) and the showers, simulated with our model (second row) for three different sets of input parameters. Color represents $log_{10}(\frac{E}{MeV})$ for every cell.

5 Experiments

We start with comparing original clusters, produced by full `Geant4` simulation and clusters generated by the trained model for the same parameters of the incident particles: the same energy, the same direction, and the same position on the calorimeter face. Corresponding images for four arbitrary parameter sets are presented in Fig. 3. These images demonstrate the very good visual similarity between simulated and generated clusters.

Then we continue with a quantitative evaluation of the proposed simulation method. While generic evaluation methods for generative models exist, here we base our evaluation on physics-driven similarity metrics. These metrics are designed using the domain knowledge and the recommendations from physicists on the evaluation of simulation procedures. For this presentation, we selected a few cluster properties which essentially drive cluster properties used in the reconstruction of calorimeter objects and following physics analysis. If the initial particle direction is not perpendicular to the calorimeter face, the produced cluster is elongated in that direction. Therefore, we consider separately cluster width in the direction of



(a) The transverse width of real and generated clusters

(b) The longitudinal width of real and generated clusters

(c) ΔX between cluster center of mass and the true particle coordinate

(d) The sparsity of real and generated clusters

(e) The transverse asymmetry of real and generated clusters

(f) The longitudinal asymmetry of real and generated clusters

Figure 4: Generated images quality evaluation including described physical characteristics.

the initial particle and in the transverse direction. Spatial resolution, which is the distance between the centre mass of the cluster and the initial track projection to the shower max depth, is another important characteristic affecting the physics properties of the cluster. Cluster sparsity, which is the fraction of cells with energies above some threshold, reflects the marginal low energy properties of the generated clusters. Finally, longitudinal and transverse asymmetries, which are differences in energies between forward-backwards and left-right sides of the cluster, characterise coherent energy variations. A comparison of these characteristics is presented in Fig. 4.

The primary cluster characteristics demonstrate good agreement with fully simulated data. However, secondary characteristics driven by long-range correlations between different cluster contributions might be significantly improved.

As for model performance, we trained our model for 3000 epochs which take about 70 hours on GPU NVIDIA Tesla K80. The sampling rate is 0.07 ms per sample on GPU, 4.9 ms per sample on CPU.

6 Conclusion and outlook

The research proves that Generative Adversarial Networks are a good candidate for fast simulation of high granularity detectors typically studied for the next generation accelerators. We have successfully generated images of shower energy deposition with a condition on the particle parameters, such as the momentum and the coordinates, using modern generative deep neural network techniques such as Wasserstein GAN with gradient penalty.

Future work will be focused on improving reproduction of second-order cluster characteristics, such as variations and long-range correlations between different cells.

The research leading to these results has received funding from the Russian Science Foundation under agreement No 19-71-30020.

References

[1] C. Bozzi, Tech. rep., CERN-LHCb-PUB-2015-004 (2014)

[2] J. Flynn, Tech. rep., CERN-RRB-2015-117 (2015)

[3] G. Grindhammer, S. Peters, arXiv preprint hep-ex/0001020 (2000)

[4] C. ATLAS, S. Yamamoto, M. Shapiro, J. Virzi, M. Werner, M. Venturi, M. Beckingham, E. Schmidt, T. Yamanaka, M. Duehrssen et al., Tech. rep., ATL-COM-PHYS-2010-838 (2010)

[5] M. Paganini, L. de Oliveira, B. Nachman, arXiv preprint arXiv:1705.02355 (2017)

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, *Generative adversarial nets*, in *Advances in neural information processing systems* (2014), pp. 2672–2680

[7] A. Radford, L. Metz, S. Chintala, arXiv preprint arXiv:1511.06434 (2015)

[8] P. Isola, J. Zhu, T. Zhou, A.A. Efros, arXiv preprint arXiv:1611.07004 (2016)

[9] J.Y. Zhu, T. Park, P. Isola, A.A. Efros, *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*, in *Computer Vision (ICCV), 2017 IEEE International Conference on* (2017)

[10] T.C. Wang, M.Y. Liu, J.Y. Zhu, G. Liu, A. Tao, J. Kautz, B. Catanzaro, arXiv preprint arXiv:1808.06601 (2018)

[11] M. Arjovsky, S. Chintala, L. Bottou, arXiv preprint arXiv:1701.07875 (2017)

[12] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A.C. Courville, *Improved training of wasserstein gans*, in *Advances in Neural Information Processing Systems* (2017), pp. 5769–5779

[13] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, *Deepface: Closing the gap to human-level performance in face verification*, in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2014), pp. 1701–1708

[14] L. de Oliveira, M. Paganini, B. Nachman, arXiv preprint arXiv:1701.05927 (2017)

[15] A.A. Alves Jr. et al. (LHCb collaboration), JINST **3**, S08005 (2008)

[16] J. Johnson, A. Alahi, L. Fei-Fei, *Perceptual losses for real-time style transfer and super-resolution*, in *European Conference on Computer Vision* (Springer, 2016), pp. 694–711