

Fast Simulation of Gamma/Proton Event Images for the TAIGA-IACT Experiment using Generative Adversarial Networks

Julia Dubenskaya*, Alexander Kryukov, Andrey Demichev

Lomonosov Moscow State University, SINP,
Moscow 119991, Russia

E-mail: jdubenskaya@gmail.com

High energy cosmic rays and gamma rays interacting the atmosphere produce extensive air showers (EAS) of secondary particles emitting Cherenkov light. Being detected with a telescope this light forms "images" of the air shower. In the TAIGA project, model data are widely used to analyze images obtained experimentally. The difficulty is that the computational models of the underlying physical processes are very resource intensive, since they track the type, energy, position and direction of all secondary particles born in EAS. For some data analysis purposes, this information is redundant, so that less complex and more efficient generation methods can be used. We applied a machine learning technique called Generative Adversarial Networks (GAN) to quickly generate images of two types: from gamma and protons events. As a training set, we used a sample of 2D images obtained using TAIGA Monte Carlo simulation software, containing about 50,000 events. It has been experimentally established that the generation results best fit the training set in the case when for two different types of events we create two different networks and train them separately. For gamma events a discriminator with a minimum number of convolutional layers was required, while for proton events, more stable and high-quality results are obtained if two additional fully connected layers are added to the discriminator. Testing the generators of both networks using third-party software showed that more than 90% of the generated images were found to be correct. Thus, the use of GAN provides reasonably fast and accurate simulations for the TAIGA project.

37th International Cosmic Ray Conference (ICRC 2021)
July 12th – 23rd, 2021
Online – Berlin, Germany

*Presenter

1. Introduction

High energy cosmic rays and gamma rays interacting the atmosphere produce extensive air showers (EAS) of secondary particles. The secondary particles, in turn, emit flashes of light in forward direction due to the Cherenkov effect. This light can be detected on the ground using arrays of Imaging Atmospheric Cherenkov Telescopes (IACTs) [1]. During operation, each IACT records a set of images of the air shower against the background light of the night sky.

The TAIGA experiment (Tunka Advanced Instrument for cosmic ray physics and Gamma Astronomy) [2] consists of different detector systems and measures air showers, which are initiated by charged cosmic rays or high energy gamma rays. The TAIGA Cherenkov telescope array (the TAIGA-IACT detector) is used for gamma astronomy. Gamma events are events of interest because photons, being electrically neutral, can be accurately extrapolated back to their origin. However, gamma rays of very high energies account for only a minor part of the recorded events [3], while the rest are electrically charged cosmic rays, mostly protons. Thus, in the TAIGA-IACT project, proton events are considered as background ones, and the main task is to extract rare gamma events from the proton flux.

In the TAIGA project, simulated data are widely used: for calibrating the detector, for reconstruction of air-showers and for ensuring correct gamma event/proton event separation. Thus, for the experiment to work, a large number of new simulated images of both gamma events and proton events are regularly required.

Currently, events images for the TAIGA-IACT project are modeled using special software that performs realistic Monte-Carlo simulations [4]. First, the shower itself is simulated using the CORSIKA toolkit [5], that performs detailed direct simulation of EAS evolution. The response of the IACT system was simulated using the special OPTICA-TAIGA software [6] developed at JINR, Dubna, that performs a full ray tracing of the Cherenkov photons through the telescopes' optics.

The difficulty is that the computational models of the underlying physical processes are very resource intensive and time-consuming. For some analysis purposes such as synthetic minority oversampling [7], this information is redundant, so that less complex and more efficient generation methods can be used. Here we demonstrate the possibility of using a machine learning technique called Generative Adversarial Networks (GAN) [8] to quickly generate images of gamma and protons events for the TAIGA-IACT project. The peculiarity of GANs is that they can learn from images from a given set of training data, and then generate new images that are statistically indistinguishable from the images of the training set. And although learning can take a long time, generation is very fast.

Recently, generative adversarial networks have been actively used in various fields of astrophysics. GANs were applied to generate astronomical images of galaxies [9, 10], to improve noisy astrophysical images [11, 12], to perform simulations of cosmic web [13]. Thus, the use of GANs has proven to be effective for scientific applications in astrophysics. Further in this paper, we will show the results of applying this method to generate images of both gamma and proton events of the Cherenkov telescope for the TAIGA-IACT project.

The rest of the article is structured as follows. In the second section, we briefly review the basic idea of a GAN in general. In the third section we describe how the data for the training set

was prepared in our case. In the fourth section, we present the network architecture for generating gamma and proton events. In the fifth section the correctness of the results obtained is assessed. The final section contains a brief summary and a discussion of the results.

2. Generative Adversarial Networks

From a technical point of view a GAN [8] is a special class of machine learning frameworks designed to generate, improve or process images. Each GAN consists of two contesting parts: a generator and a discriminator, and each part itself is a neural network.

Generator is a neural network that takes a random vector as input and tries to transform this input into images similar to the real ones. Generator performs the transpose convolution procedure thus increasing the size of its output relative to the input. Discriminator is also a neural network that tries to distinguish between real images and fake images produced by the generator. In fact, discriminator is a classic convolutional neural network (CNN) that performs image classification. Generator and discriminator are trained together on real images in an adversarial game, until the discriminator model is fooled about half the time, meaning the generator model is generating plausible examples. The structure of a typical GAN is shown in Figure 1.

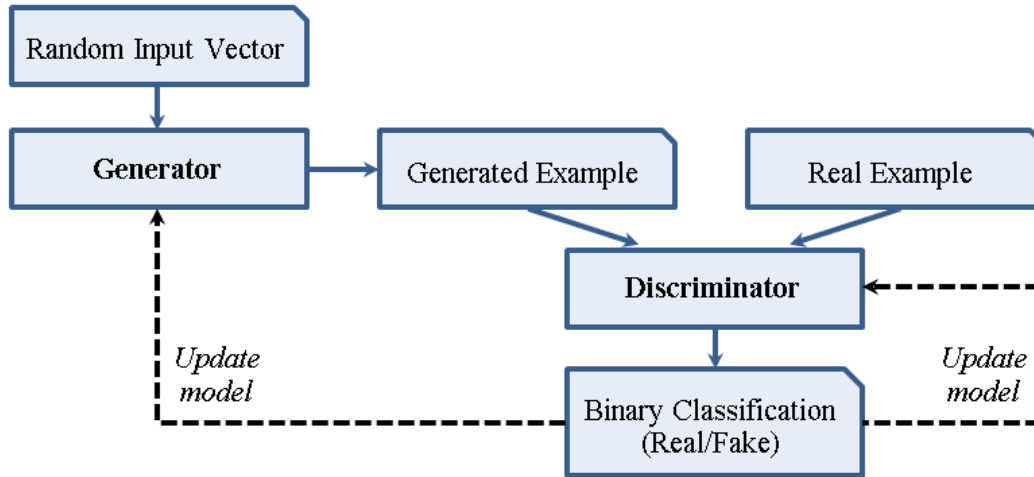


Figure 1. The structure of a typical GAN

In addition to the trainable network parameters, there are also hyperparameters (number of inner layers, filter size for each layer, activation functions, etc.) that are selected manually for each specific task. We have considered different combinations of hyperparameters and further in the article we suggest the combination that provided us with the best results for today.

3. Training set preprocessing

For training we used a sample of two-dimensional images obtained using OPTICA-TAIGA Monte Carlo simulation software, containing 25000 gamma events and 25000 proton events. It is worth noting that preprocessing of the training set is extremely important for generating images similar to those recorded by IACT. When preparing the training sample images, we applied cleaning, coordinate transformation, image resizing and pixel values recalculation.

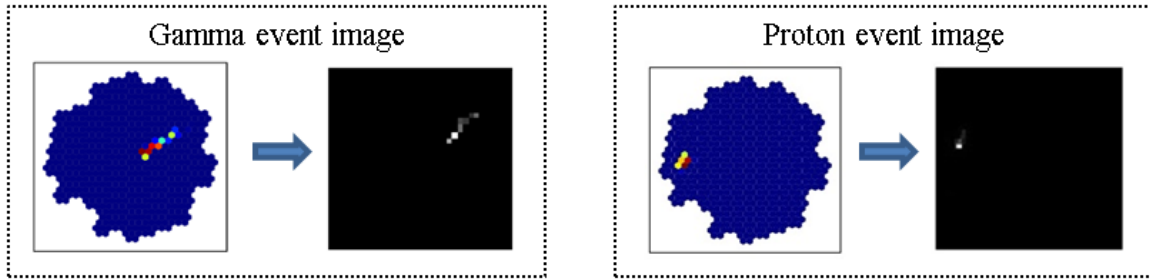


Figure 2. Examples of the input images

Examples of the original images and the images after preprocessing are shown in Figure 2.

3.1 Cleaning

Normally, Monte Carlo images contain noise from the night sky background fluctuations and electronic components. Image cleaning is a conventional procedure to remove such noise thus leaving only images produced by a shower of secondary particles [14]. In theory, GANs can generate noisy images as well as cleaned ones. Our experiments show that our GAN reproduces cleaned images better. In order to teach our GAN to generate cleaned images, we had to clean the images from the training set.

3.2 Coordinate transformation and image resizing

The TAIGA-IACT telescope cameras consist of arrays of photomultipliers arranged in a hexagonal grid. Each photomultiplier produces one pixel of the image. Accordingly, the original images are also hexagonal. However, the current high performance GAN implementations are designed for square grids. That's why first, we generated rectangular images from hexagonal ones by transition to an oblique coordinate system. As a result, we got images of 31 by 30 pixels. Then, to make the images square, we resized each image to 32 by 32 pixels by adding two columns of zeros to the right and one row of zeros to the bottom.

3.3 Pixel values recalculation

Since the training set contains images with different pixel values (i.e. different energies), we had to switch to a logarithmic scale by applying the logarithm function to each pixel of each image: $\ln(1+x)$.

4. GAN architecture for the TAIGA-IACT project

It was experimentally found that training is more stable and the generation results best correspond to the training set in the case when separate training is performed for two different types of events, that is, we train one network to generate proton event images and another to generate gamma event images. In fact, these networks differ only in the architecture of the discriminators, while the architecture of the gamma image generator is identical to the architecture of the proton image generator.

4.1 Generator architecture (common for both gamma events and proton events)

This architecture of the generator (see Figure 3) is more or less standard and similar to those suggested in the books for image generation. The generator takes a random vector as input, and then uses transpose convolution to upsample until it gets an image of the desired size. All layers except the output layer use 4x4 filters and a leaky ReLU function with $\alpha=0.2$ as the activation function. The output layer has one 6x6 filter and uses a sigmoid for its activation.

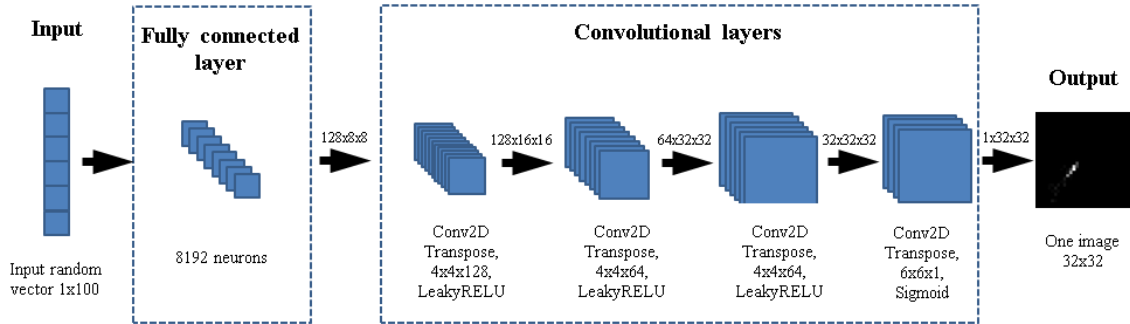


Figure 3. The generator architecture

4.2 Discriminator architecture for gamma events

The discriminator architecture for gamma (see Figure 4) is also more or less standard. The discriminator for gamma events is a small classic convolutional network that uses 4x4 filters in all layers. The convolutional layers use a leaky ReLU function with $\alpha=0.2$ as the activation function. The output layer uses a sigmoid as its activation function.

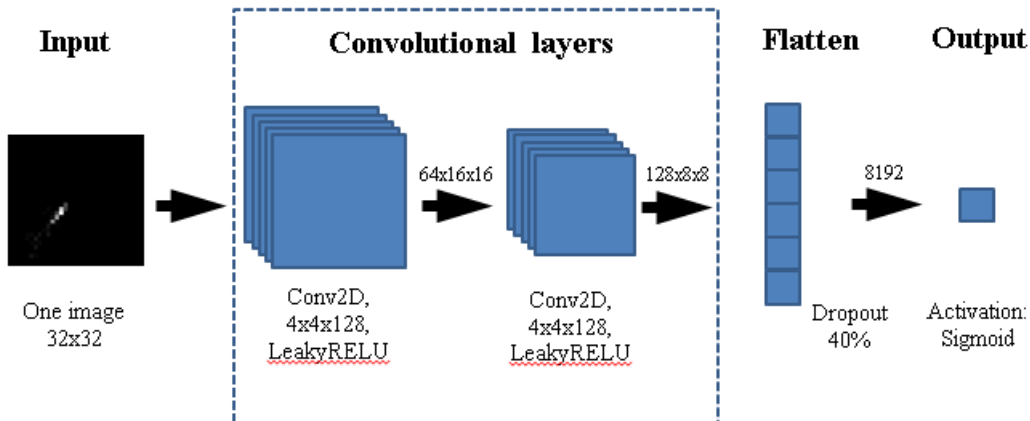


Figure 4. The discriminator architecture for gamma events

4.3 Discriminator architecture for proton events

The discriminator architecture for proton events is shown in Figure 5. We found that for proton events having a standard architecture, our discriminator becomes unstable and from time to time generates random additional spots, noise, or completely black images. Our experiments have shown that adding several dense fully connected layers before the output layer solves the problem. In the convolutional layers, the convolution filter size is 4x4; the leaky ReLU function with $\alpha=0.2$ is used for the activation. In the fully connected layers the activation function is

a leaky ReLU function with $\alpha=0.2$, and the dropout with a probability of 0.4 is also used. The output layer uses a sigmoid function for its activation.

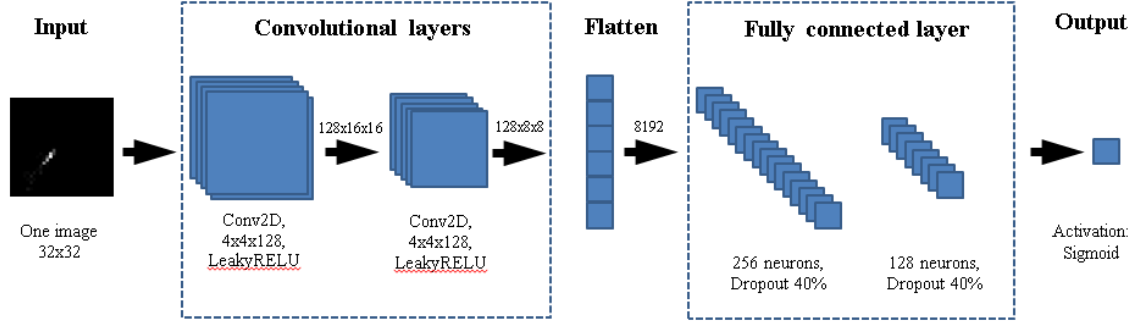


Figure 5. The discriminator architecture for proton events

5. Simulation results

Based on the above, we created two separate GANs - for gamma and for protons - and for each of them we selected 25000 events of the corresponding type as a training set. Thus, for the gamma network we used only gamma events and for the proton network we used only proton events. We implemented these networks using a TensorFlow [15] software package. Network training at the GPU Tesla P100 took about 6 hours for each network. After training, generation of 4000 events (of any type) takes about 10 seconds. Then the generated images are converted back to hexagonal form.

Examples of the generated images and the images after back conversion are shown in Figure 6.

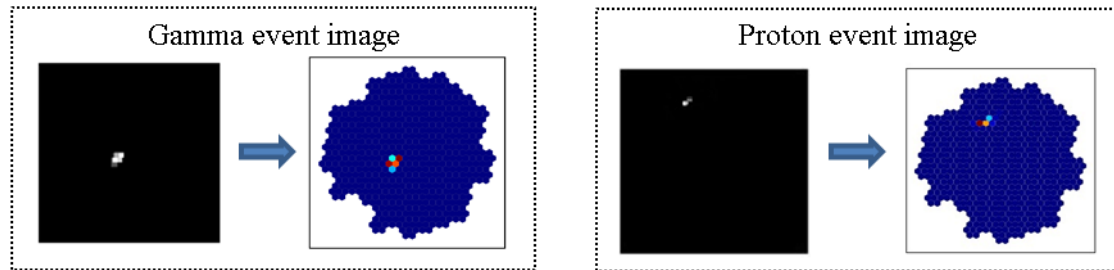


Figure 6. Examples of the generated images

For verification, we generated two samples of 4000 event images of each type and classified them using the third party software tool that is used for classification in the TAIGA-IACT project [16]. This software tool determines the probability that an image is a gamma image (gamma likelihood).

The plot in Figure 7 shows the results of the gamma classification - the distribution of the number of generated gamma events by probabilities. The X-axis in the plot represents the probability that the image is a gamma event and the Y-axis is the number of generated gamma events classified as gamma events with a given probability. The plot shows that for more than half of the generated events, the calculated probability is 90-100%. Moreover, for 94.5% of generated events, the probability exceeds 50% - thus, these events are recognized as gamma events.

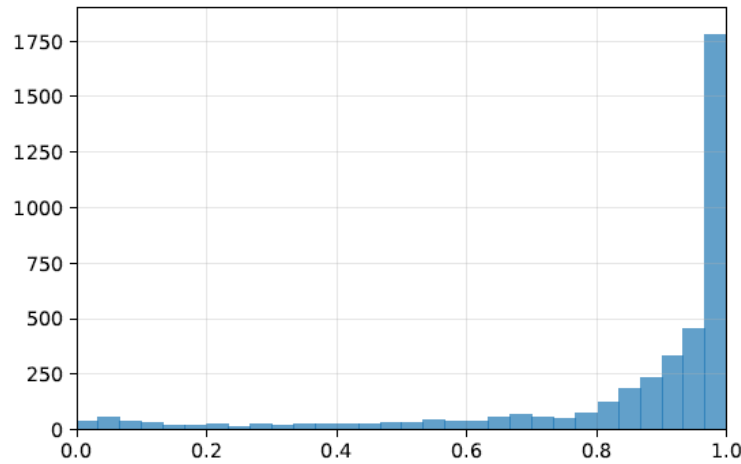


Figure 7. The gamma likelihood for the gamma events

The plot in Figure 8 shows the results of the proton classification. The X-axis in the plot represents the probability that the image is a gamma event and Y-axis is the number of generated proton events classified as gamma events with a given probability. As can be seen, for most of the generated proton events, the calculated probability is close to zero, which means that the generated proton events differ with confidence from gamma events.

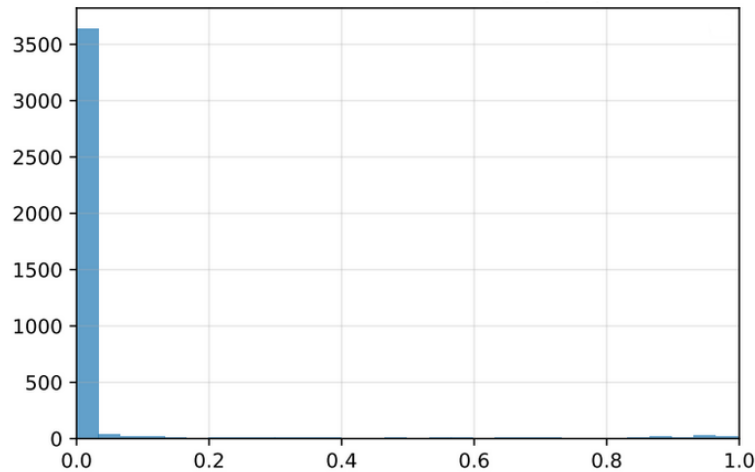


Figure 8. The gamma likelihood for the proton events

So, the verification showed that most of the generated images of both types turned out to be very similar to the images from the training set, while these are completely new images that can be used in the experiment.

6. Conclusion

Summarizing the above, we can conclude that generative adversarial networks simulate proton and gamma events for the TAIGA-IACT experiment with a high degree of accuracy and reliability. Most of the generated events are indistinguishable from the events generated using the traditional Monte Carlo method. At the same time, the rate of generation of events using GANs is much higher than the rate of generation by the Monte Carlo method. To ensure that the

generated sample contains only correct images, all generated images should be checked by the TAIGA-IACT classification program, with inappropriate images being discarded.

In our future work, we hope to further increase the accuracy of modeling, aiming, among other things, to minimize the number of inappropriate images.

References

- [1] T.C. Weekes et al., *Observation of TeV gamma rays from the Crab Nebula using the atmospheric Cherenkov imaging technique*, *Astroph. J.* **342**, (1989) 379
- [2] N. Budnev et al., *The TAIGA experiment: From cosmic-ray to gamma-ray astronomy in the Tunka valley*, *Nucl.Instrum.Meth.* **A845** (2017) 330-333
- [3] E. Lorenz, R. Wagner, *Very-high energy gamma-ray astronomy*, *EPJ H* **37**, (2012) 459–513.
- [4] M.H. Kunnas, *Simulation of imaging air shower Cherenkov telescopes as part of the TAIGA Project*, *Proc. of Magellan Workshop* (DESY-PROC-2016-05), doi: 10.3204/DESY-PROC-2016-05/33
- [5] D. Heck et al., *CORSIKA: A Monte Carlo code to simulate extensive air showers*, *Forschungszentrum Karlsruhe FZKA 6019*, (1998)
- [6] E.B. Postnikov et al., *Hybrid method for identifying mass groups of primary cosmic rays in the joint operation of IACTs and wide angle Cherenkov timing arrays*. *J.Phys.: Conf. Series* **798**, 012030 (2017)
- [7] N.V. Chawla et al., *Smote: synthetic minority over-sampling technique*, *J. Mach. Learn. Res.* (2002) 321-357.
- [8] I.J. Goodfellow et al., *Generative Adversarial Networks*, *ArXiv* 1406.2661 (2014)
- [9] J. Regier et al., *A deep generative model for astronomical images of galaxies*, *Proc. of NIPS Workshop: Advances in Approximate Bayesian Inference* (2015)
- [10] S. Ravanbakhsh et al., *Enabling dark energy science with deep generative models of galaxy images*, *AAAI* (2017) 1488–1494
- [11] K. Schawinski et al., *Generative adversarial networks recover features in astrophysical images of galaxies beyond the deconvolution limit*, *MNRAS* **467** (2017) L110–L114
- [12] Long Xu et al., *Solar Image Deconvolution by Generative Adversarial Network*, *Res. Astron. Astrophys.* **20** (2020) 170
- [13] A.C. Rodriguez et al., *Fast cosmic web simulations with generative adversarial networks*, *Comput. Astrophys.* **5**, 4 (2018)
- [14] E.B. Postnikov et al., *Particle identification in ground-based gamma-ray astronomy using convolutional neural networks*, *Proc. of Int. Conf. Distributed Computing and Grid-technologies in Science and Education* (2018) 431-435
- [15] TensorFlow Homepage, <http://www.tensorflow.org>. Last accessed 05 July 2021
- [16] E.B. Postnikov et al., *Deep Learning for Energy Estimation and Particle Identification in Gamma-ray Astronomy*. *Proc. of the 3rd Int. Workshop DLC-2019*, *CEUR-WS* **2406** (2019) 90-99