ISSN 0027-1349, Moscow University Physics Bulletin, 2024, Vol. 79, Suppl. 2, pp. S684–S689. © Allerton Press, Inc., 2024.

Generation of Grid Surface Detector Data in the Telescope Array Experiment Using Neural Networks

R. R. Fitagdinov[1,2*] and I. V. Kharuk[1,2]**

[1]*Institute for Nuclear Research, Russian Academy of Sciences, Moscow, 117312 Russia*
[2]*Moscow Institute of Physics and Technology (National Research University), Dolgoprudny, Moscow oblast, 141700 Russia*

Received October 1, 2024; revised October 10, 2024; accepted October 20, 2024

Abstract—In this article, we talk about generating data obtained in the Telescope Array experiment. For this we are using Wasserstein's generative adversarial networks. Wasserstein's generative adversarial networks were trained on data obtained using the Monte Carlo method. To improve the quality of the generation, we add the loss function for the generator, which is based on the physics of the process of spreading an extensive air shower. In the future, this network can be used to search for anomalies and for faster data generation, compared with algorithms based on the Monte Carlo method.

Keywords: telescope array, GAN, data sampling, astrophysics

DOI: 10.3103/S0027134924702138

1. INTRODUCTION

Cosmic rays are charged particles that reach the Earth from space. The sources of charged particles detected by the Telescope Array experiment can be distant space objects, such as pulsars, galactic nuclei, etc. Cosmic rays provide us with unique information about such distant space objects. Despite its name, cosmic rays (CR) are not rays in the traditional sense, but charged particles accelerated to energies above 10^8 eV by powerful astrophysical processes. Also, the energies possessed by cosmic rays can exceed the energy thresholds reached at accelerator complexes by several orders of magnitude. An example of such an event is a particle with an energy of 244 EeV recently detected in the Telescope Array experiment and named Amaterasu [1].

Upon reaching the Earth's atmosphere, charged particles interact with particles of the atmosphere, during which many secondary particles are born. In turn, these secondary particles, which still have huge energy, also interact with molecules of the atmosphere. Thus, a huge number of secondary particles reach the Earth, which are detected by the detectors of the Telescope Array experiment [2]. The cascade of secondary particles, called an extensive air shower (EAS), has a stochastic character, due to the large number of interactions of secondary particles in the atmosphere. The EAS is researched in large cosmic ray observatories, such as Telescope Array [2], Pierre Auger Observatory [3], LHAASO [4], and others.

Data generated by models based on the Monte Carlo method is used to analyze the EAS. High-energy interactions are modeled using the following programs: SIBYLL [5], QGSJET-II-03 [6], QGSJET II-04 [7], and others. In our work, we applied the data obtained using the last two models. However, this method has several difficulties. The first and most important thing is that the models of particle behavior are not accurate. The most famous witness to this discrepancy is the Muon Deficit [8].

A generative adversarial network (GAN) [9] and anomaly search algorithms [10] based on them can help to analyze the discrepancy between real data and generated by the Monte Carlo method. GANs also work several orders of magnitude faster than methods based on the Monte Carlo method. The similar work for the Ice Cube experiment showed which GANS can be used for simulated physics experiments [11].

In this article, we report on the generation of the surface detector grid readings on the Telescope Array experiment. For this tasks, we used Wasserstein's generative adversarial network (WGAN) with gradient penalty [12]. Also we wrote the loss function

*E-mail: robertfitagdinov@gmail.com
**E-mail: ivan.kharuk@phystech.edu

based on the physics of the propagation of an extensive air shower.

2. DATA AND METHOD

2.1. Telescope Array Experiment

Telescope Array is the largest experiment detecting cosmic rays in the northern hemisphere [2]. The experiment is located in the desert of Utah in the United States. It is designed to work with ultra-high energy cosmic rays with an energy of about $10^{17.5}$ eV or more [13]. The experiment consists of 507 surface scintillation detectors and 3 fluorescent detectors located at three points around the field of scintillation detectors.

In our work, we used data only from scintillation detectors obtained using Monte Carlo simulation. Each detector has an area of 3 m^2. The detectors form a square grid with an average step of about 1.2 km. In total, all 507 surface scintillation detectors cover an area of 762 km^2. Each detector consists of 2 layers of a 1.2 cm thick scintillator each.

2.2. Data

The data for training the neural network were generated using the Monte Carlo method. In the simulation, we used CORSICA [14] and high energy hadron interaction models QGSjetII-03 [6]. In total, we used 472 351 Monte Carlo simulated events.

We worked with a grid of **6X6** detectors, which makes a total of 36 detectors. After reconstruction of the EAS propagation axis [15], a **6X6** detector grid is built so that the center of the grid is as close as possible to the center of the shower (the intersection point of the axis of the shower and the Earth's surface). The size of the detector grid was chosen as the most appropriate: events in which all triggered detectors lie in this grid account for 99.96% of all events.

Each detector has 4 channels of data:

- The integral signal of the detector, measured in minimal ionization partial (MIP) units [15].
- The arrival time of the reconstructed flat EAS front.
- The difference between the time of registration by the EAS detector and the time of arrival of the reconstructed flat front.
- Mask indicating whether the detector has been triggered. The mask takes discrete values: 0 means the detector did not work, or 1 means the detector worked.

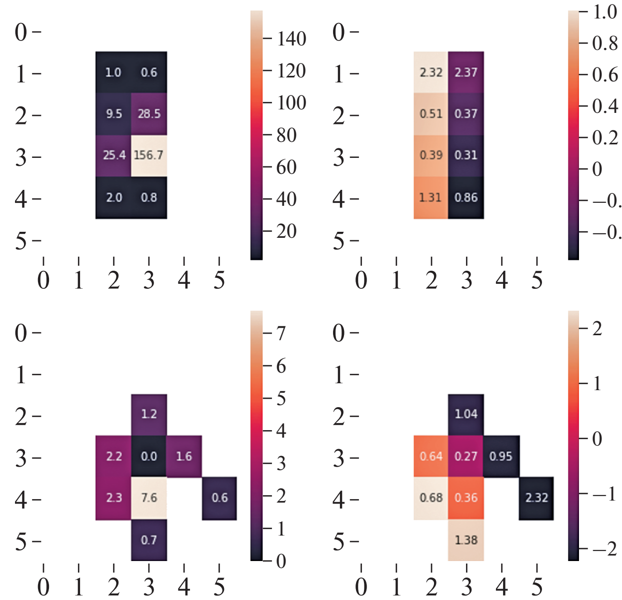

Fig. 1. Example of data.

An example of the data can be seen in Fig. 1. It shows an example of 2 events of the detection of EAS (top and bottom). The left part of the figure shows the integral signal registered by the detector (color value is equal to numerical value). The right part shows in the color channel the time of arrival of the reconstructed flat front, and in the numerical channel—the real time of detection by each detector. Empty cells are failed detectors in the event (having a mask equal to 0).

2.3. WGAN's Architecture

GANs consist of two neural networks, a generator and a discriminator, that are trained simultaneously in a competitive setting [9]. The generator creates synthetic data samples from random noise, aiming to produce outputs indistinguishable from real data. The discriminator, on the other hand, evaluates both real and generated samples, providing feedback on their authenticity. The generator improves its output based on the discriminators feedback, while the discriminator enhances its ability to differentiate between real and fake data. This adversarial process continues until the generator produces high-quality samples that the discriminator can no longer reliably distinguish from real data, leading to successful generative modeling.

WGAN-GP, or Wasserstein generative adversarial network with gradient penalty [12], is an advanced GAN architecture that improves training stability and mitigates issues like mode collapse by using the Wasserstein distance as a loss metric. Instead of a traditional discriminator, it employs a critic that scores both real and generated samples, maximizing the difference in these scores. In this paper, we will write WGAN, meaning that we use WGAN-GP.

As a preprocessing, we normalized the data. The data was normalized in such a way that the values of each channel were in the range from 0 to 1. So the normalization for n channel can be written as: $d_n = (d_n - d_{\min})/(d_{\max} - d_{\min})$, where $d_{\min} = \min(d_n)$; $d_{\max} = \max(d_n)$.

The critic model had 4 convolutional layers, which increased the number of detector channels to 64, which allows for a greater variety of features transmitted to subsequent fully connected layers. We applied the GELU [16] activation function between the layers. After the last convolutional layer and the reversal, the output tensor has the shape **(576,)**. In the next step, the tensor inputs in 2 consecutive fully connected layers with output dimensions of tensors 30 and 1, respectively. So the critic model had about 40 thousand trainable parameters.

Transposed and classical convolutional layers were used in the generator model. The model received a Gaussian noise input of dimension 50. After that, the noise was transformed by a fully connected layer into a 512-dimensional tensor. Then the tensor was expanded to the dimension (4,4,32) without changing the numerical values of the tensor, using the reshape function. After that, the tensor passed through a convolutional layer with a core of dimension 1 and channel dimension 32. This layer helped to bring the parameters of the transmitted data into a format convenient for further training. After that, 3 convolutional layers were applied sequentially, which transformed the signal into a tensor with dimension (32,32,32). In the next step, the tensor lowered the spatial dimension to (16.16) using the AveragePooling layer. The logic of its application is similar to its application in the task of generating the signal of the most active detector in an event. After that, the tensor passed through 2 convolutional layers. Between those layers, we also applied the AveragePooling layer. So the output tensor had a spatial dimension equal to (8.8). At the final stage, to obtain the necessary dimension in (6,6), we discarded 1 value on each side of the tensor using a two-dimensional Cropping layer. Thus, the generator had about 54 thousand trainable parameters.

The scheme of the (a) critic and (b) generator can be seen in the Fig. 2.

The output of the WGAN can be seen in Fig. 3. WGAN was able to correctly reproduce the values of the integral signal. However, it generated the times of EAS arrival with errors. They are not consistent with the physics of the process. The time of the flat front is not linear, and the curvature of the front is not uniform from the axis of the shower.

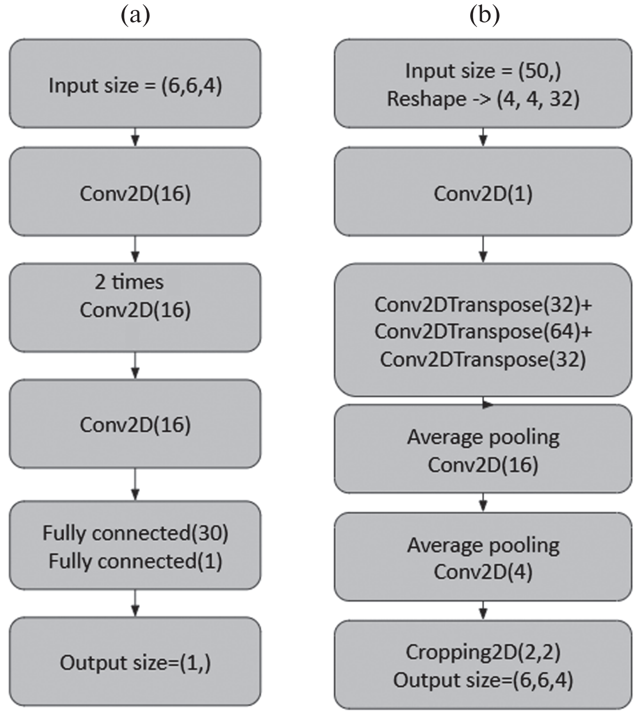

Fig. 2. The scheme of the critic (a) and generator (b).

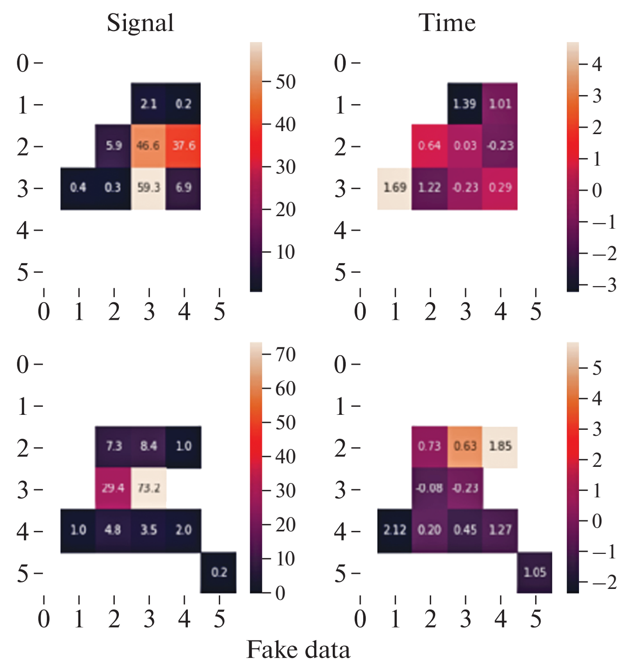

Fig. 3. The output of the WGAN with the original loss.

2.4. Additional Loss Function

For the generator to get data consistent with the physics of the EAS propagation process, we wrote the additional loss function. This loss function was added to the generator loss in the training of the WGAN. Thus, the generator is punished for implausible data not only by the critic, who may find it difficult to grasp the physics of the process but also by the prescribed physics of the propagation of the EAS. Thus, the loss of the generator in training will look like:

$$L_{\text{gen}} = L_{\text{orig}} + WL_{\text{add}}, \tag{1}$$

where L_{orig}—generator's loss in paper [12]. L_{add}—our add loss which added with weight W. In our work we define $W = 1$.

The chi-squared likelihood of fitting the activation time of the detectors was chosen as this loss function. The reconstruction procedure described in the article [15] was chosen as the citation of the signal.

To reconstruct the time of the detection EAS, the algorithm searches for the most optimal shower parameters. The whole process is described by 7 values:

1. Zenith angle θ.
2. Azimuth angle ϕ.
3. Curvature of the front (Linsley front curvature) a.
4. Function of the spatial distribution of the signal at a distance of 800 m from the center of the shower S_{800}.
5. X coordinate of the intersection of the axis of the shower with the surface of the Earth $\vec{R}_{\text{core}}\,(x)$.
6. Y coordinate of the intersection of the axis of the shower with the surface of the Earth $\vec{R}_{\text{core}}\,(y)$.
7. Value t_0 is the offset constant of a fixed signal according to the formula (2).

The time of the detection of EAS is reconstructed as

$$t(\vec{R}) = t_0 + t_{\text{plane}}\,(\vec{R}) + a \times (1 + r/R_L)^{1.5} \times LDF(r)^{-0.5}, \tag{2}$$

where

$$t_{\text{plane}}\,(\vec{R}) = \frac{1}{c}\vec{n}\left(\vec{R} - \vec{R}_{\text{core}}\right),$$

where $\vec{n}$ is the EAS propagation vector depending on the angles θ and ϕ. $\vec{R}_{\text{core}} = (\vec{R}_{\text{core}}(x); \vec{R}_{\text{core}}(y); 0)$. Thus, the second term in the equation 2 describes the arrival time of the reconstructed flat front. $t_{\text{plane}}(\vec{R})$ is the time of the reconstructed flat EAS front at the $\vec{R}$.

The third term in the equation (2) describes the curvature of the real EAS front. It depends on the curvature of the front (a), the distance of the detector to the shower axis (r), and the value of LDF, which is written as:

$$LDF(r) = \left(\frac{r}{R_m}\right)^{-1.2} \times \left(1 + \frac{r}{R_m}\right)^{-(\eta-1.2)} \times \left(1 + \frac{r^2}{R_1^2}\right)^{-0.6}, \tag{3}$$

where

$$r = \sqrt{\left(\vec{R} - \vec{R}_{\text{core}}\right)^2 - \left(\vec{n}\left(\vec{R} - \vec{R}_{\text{core}}\right)\right)^2},$$
$$R_m = 90.0 \text{ m}, R_1 = 1000 \text{ m}, R_L = 30 \text{ m},$$
$$\eta = 3.97 - 1.79(\sec(\theta) - 1).$$

In this equation LDF is the empirical lateral distribution profile [15].

The measurement error used in the chi-squared calculation was considered as:

$$T_{\text{err}} = t_d(S_{800})^{0.2}, \tag{4}$$

where t_d is the third term in the equation (2) which describes the curvature of the EAS front.

The algorithm calculates the chi-squared value, which shows how close the values of the detection time of EAS are to the times obtained using the equation (2). The algorithm is optimized by the chi-squared value, looking for the optimal parameters described above. Thus, at the output, we get the optimal values of the EAS parameters. Thus, the chi-squared value, acting as the value of the loss function, helps the generator to learn how to get more realistic results and penalizes if the data does not correspond to the physics of the process, such as, for example, a limit on the speed of signal propagation.

By applying the reconstruction algorithm to our data obtained using the Monte Carlo method, we were able to obtain parameters close to those set during sampling only for high zenith angles. The results of the reconstruction can be seen in the Fig. 4.

The figure shows the dependencies of the real angles (parameters when setting the Monte Carlo simulation) along the Y-axis and their values obtained by reconstruction along the X-axis. The results for zenith angles are shown on the left and for azimuth angles on the right. As you can see, events with a real zenith angle of less than 20° (almost vertical EAS drop) are poorly reconstructed.

The reason for the inability to accurately quote the parameters of the EAS at low zenith angles may be the incorrect behavior of the optimizer used in these cases.

Due to the inability to solve this problem, it was decided to work with events having zenith angles of more than 20°. The result of their reconstruction can be seen in Fig. 4 in green.

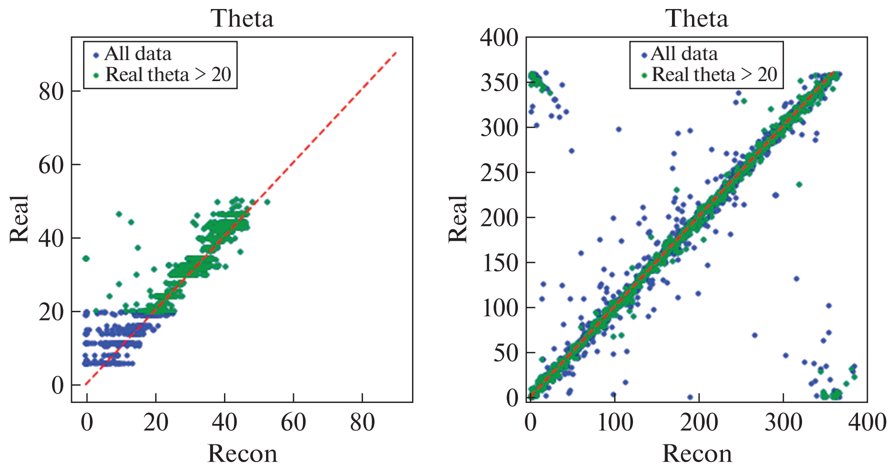

Fig. 4. Reconstruction of angles. The zenith angles are shown on the left side, and the azimuth angles are on the right. Blue—reconstruction of all events, green—events with a real zenith angle of more than 20^{circ}.

The points on the right side of Fig. 4 responsible for the azimuth angle, which are located in the upper left and lower right corners of the graph, are not as incorrect as they may seem. In all calculations, it is not so much the azimuthal angle that is important to us, as the trigonometric functions from it. The azimuth angle in the formulas (2) and (3) is used only to specify the vector of the shower axis $\vec{n}$. It can be written as:

$$\vec{n} = (\sin(\theta)\sin(\phi); \sin(\theta)\cos(\phi); \cos(\theta)). \quad (5)$$

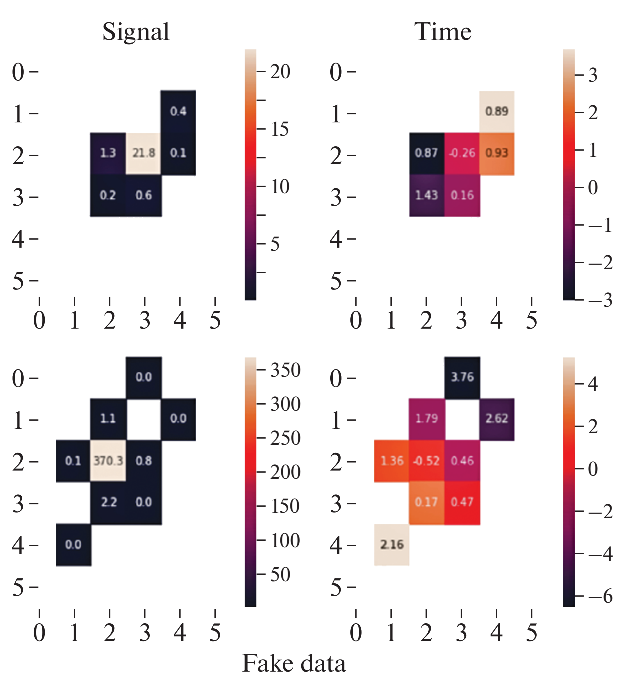

Fig. 5. The output of the WGAN with the additional physical loss function.

3. RESULTS AND CONCLUSIONS

Within the framework of the presented work, the WGAN was written to generate an integrated signal and the arrival time of the shower for the 6X6 detector grid. For better data generation, an additional loss function based on the physics of the EAS propagation process was written, which improved the generation results.

The result of the WGAN with the added loss function can be seen in Fig. 5. As can be seen, the time of the reconstructed flat front has become more linear. Also, the curvature of the front, which can be seen from the numerical channel in the right images, has become more uniform in each event. Thus, using the additional loss function when training our GAN results in the generated data better matching the physics of the process.

Thus, the data generated by WGAN had a chi-squared probability error value less than before our loss function was enabled (it was **4.2** on average, it became **2.1**). However, this loss function could only be applied to events with the zenith angle more than 20°.

In further work, we plan to solve the problem of reconstructing the parameters of the EAS for small zenith angles. Thanks to this, it will be possible to use a trained WGAN to generate the entire dataset. As mentioned above, such generative networks can be used to search for anomalies in the data obtained using Monte Carlo.

FUNDING

The work was supported by the Russian Science Foundation, grant no. 22-22-20063.

CONFLICT OF INTEREST

The authors of this work declare that they have no conflicts of interest.

REFERENCES

1. R. G. Lang, arXiv Preprint (2024). https://doi.org/10.48550/arXiv.2405.03528
2. R. U. Abbasi, M. G. Allen, R. Arimura, J. W. Belz, D. R. Bergman, S. A. Blake, B. K. Shin, I. J. Buckland, B. G. Cheon, T. Fujii, K. Fujisue, K. Fujita, M. Fukushima, G. D. Furlich, Z. R. Gerber, N. Globus, K. Hibino, R. Higuchi, K. Honda, D. Ikeda, H. Ito, A. Iwasaki, S. Jeong, H. M. Jeong, C. H. Jui, K. Kadota, F. Kakimoto, O. E. Kalashev, K. Kasahara, K. Kawata, I. Kharuk, E. Kido, S. W. Kim, H. B. Kim, J. H. Kim, J. H. Kim, I. Komae, Y. Kubota, M. Y. Kuznetsov, K. H. Lee, B. K. Lubsandorzhiev, et al. (Telescope Array Collaboration), Science **382**, 903 (2023). https://doi.org/10.1126/science.abo5095
3. P. Abraham, M. Abreu, C. Aglietta, Aguirre, D. Ej Ahn, I. Allard, J. Allekotte, P. Allen, J. Allison, Alvarez-Muniz, and Others, Nucl. Instrum. Methods Phys. Res., Sect. A **620**, 227 (2010). https://doi.org/10.1016/j.nima.2010.04.023
4. H. He (for the LHAASO Collaboration), Radiation Detection Technology and Methods **2**, 7 (2018). https://doi.org/10.1007/s41605-018-0037-3
5. F. Riehn, H. P. Dembinski, R. Engel, A. Fedynitch, T. K. Gaisser, and T. Stanev, arXiv Preprint (2017). https://doi.org/10.48550/arXiv.1709.07227
6. L. G. Dedenko, A. V. Lukyashin, T. M. Roganova, and G. F. Fedorova, EPJ Web Conf. **158**, 06006 (2017).
7. N. Nesterova, arXiv Preprint (2018). https://doi.org/10.48550/arXiv.1802.07357
8. F. Gesualdi, A. D. Supanitsky, and A. Etchegoyen, Phys. Rev. D **101**, 083025 (2020). https://doi.org/10.1103/PhysRevD.101.083025
9. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, Sh. Ozair, A. Courville, and Yo. Bengio, Commun. ACM **63**, 139 (2020). https://doi.org/10.1145/3422622
10. T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, in *Information Processing in Medical Imaging. IPMI 2017*, Ed. by M. Niethammer et al., Lecture Notes in Computer Science, Vol. 10265 (Cham, 2017), p. 146. https://doi.org/10.1007/978-3-319-59050-9_12
11. A. Holmberg, "Fast simulations of radio neutrino detectors: Using generative adversarial networks and artificial neural networks," MSc Thesis (Uppsala University, Uppsala, Sweden, 2022).
12. I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, arXiv Preprint (2017). https://doi.org/10.48550/arXiv.1704.00028
13. T. Abu-Zayyad, R. Aida, M. Allen, R. Anderson, R. Azuma, E. Barcikowski, J. W. Belz, D. R. Bergman, S. A. Blake, R. Cady, B. G. Cheon, J. Chiba, M. Chikawa, E. J. Cho, W. R. Cho, H. Fujii, T. Fujii, T. Fukuda, M. Fukushima, D. Gorbunov, W. Hanlon, K. Hayashi, Y. Hayashi, N. Hayashida, K. Hibino, K. Hiyama, K. Honda, T. Iguchi, D. Ikeda, K. Ikuta, N. Inoue, T. Ishii, R. Ishimori, D. Ivanov, S. Iwamoto, C. C. H. Jui, K. Kadota, F. Kakimoto, O. Kalashev, T. Kanbe, K. Kasahara, H. Kawai, S. Kawakami, S. Kawana, E. Kido, H. B. Kim, H. K. Kim, J. H. Kim, J. H. Kim, et al., Nucl. Instrum. Methods Phys. Res., Sect. A **689**, 87 (2012). https://doi.org/10.1016/j.nima.2012.05.079
14. D. Heck, J. Knapp, J. N. Capdevielle, G. Schatz, and T. Thouw, CORSIKA: A Monte Carlo code to simulate extensive air showers, Technical Report FZKA-6019 (Forschungszentrum Karlsruhe, Karlsruhe, 1998). https://digbib.bibliothek.kit.edu/volltexte/fzk/60-19/6019.pdf.
15. D. Ivanov, O. E. Kalashev, M. Yu. Kuznetsov, G. I. Rubtsov, T. Sako, Y. Tsunesada, and Y. V. Zhezher, Mach. Learn.: Sci. Technol. **2**, 015006 (2020). https://doi.org/10.1088/2632-2153/abae74
16. D. Hendrycks and K. Gimpel, arXiv Preprint (2016). https://doi.org/10.48550/arXiv.1606.08415

Publisher's Note. Allerton Press, Inc. remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

AI tools may have been used in the translation or editing of this article.