NUCLEAR PHYSICS

Original article
DOI: https://doi.org/10.18721/JPM.16414

A GENERATOR OF DEEP INELASTIC LEPTON-PROTON SCATTERING BASED ON THE GENERATIVE-ADVERSARIAL NETWORK (GAN)

A. A. Lobanov✉, *Ya. A. Berdnikov*

Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russia

✉ lobanov2.aa@edu.spbstu.ru

Abstract. The paper considers the application of a Generative Adversarial Network (GAN) for the development of a generator of deep inelastic lepton-proton scattering. The difficulty of effective training of the generator based on GAN is noted. It is associated with the use of complex schemes of distributions of physical properties (energies, momentum components, etc.) of particles in the process of deeply inelastic lepton-proton scattering. It is shown that the GAN makes it possible to faithfully reproduce the distributions of lepton physical properties in the final state at different initial energies of the center of mass in the range between 20 and 100 GeV.

Keywords: inclusive deep inelastic scattering, neural network, generative adversarial network, lepton-proton scattering

Citation: Lobanov A. A., Berdnikov Ya. A., A generator of deep inelastic lepton-proton scattering based on the Generative-Adversarial Network (GAN), St. Petersburg State Polytechnical University Journal. Physics and Mathematics. 16 (4) (2023) 181–188. DOI: https://doi.org/10.18721/JPM.16415

Научная статья
УДК 539.12
DOI: https://doi.org/10.18721/JPM.16414

ГЕНЕРАТОР ГЛУБОКО НЕУПРУГОГО РАССЕЯНИЯ ЛЕПТОНОВ НА ПРОТОНЕ НА ОСНОВЕ ГЕНЕРАТИВНО-СОСТЯЗАТЕЛЬНОЙ НЕЙРОННОЙ СЕТИ

А. А. Лобанов✉, *Я. А. Бердников*

Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия

✉ lobanov2.aa@edu.spbstu.ru

Аннотация. В работе рассмотрено применение генеративно-состязательной сети (ГСС) для создания генератора глубоко неупругого лептон-протонного рассеяния. Отмечена сложность эффективного обучения генератора на основе ГСС, которая связана с использованием сложных схем распределения физических характеристик (энергий, компонентов импульсов и т. п.) частиц в процессе глубоко неупругого лептон-протонного рассеяния. Показано, что ГСС позволяет точно воспроизводить распределения физических характеристик лептона в конечном состоянии.

Ключевые слова: инклюзивное глубоко неупругое рассеяние, нейронная сеть, генеративно-состязательная сеть, лептон-протонное рассеяние

Ссылка для цитирования: Лобанов А. А., Бердников Я. А. Генератор глубоко неупругого рассеяния лептонов на протоне на основе генеративно-состязательной нейронной сети // Научно-технические ведомости СПбГПУ. Физико-математические науки. 2023. Т. 16. № 4. С. 181–188. DOI: https://doi.org/10.18721/JPM.16414

Introduction

The results of experimental studies of deep inelastic lepton-proton scattering are generally processed and analyzed by modeling both the actual process of particle interaction and the operation of detector setups; the Monte Carlo method is the most convenient for this purpose. The problem is that simulation involves complex physical models, requiring high computational costs and much time.

Machine learning methods provide an alternative, allowing to build event generators. The advantage of these methods is that they can be trained on heterogeneous data, i.e., both experimental results and data obtained by modeling the entire process under consideration (for example, inclusive deep inelastic scattering). The resulting event generator can be capable to collect the necessary data quickly and with minimal computational costs.

In this paper, we consider one of these machine learning models, the generative-adversarial network (GAN) [1].

The advantage of the considered model is its ability to faithfully reproduce the real data on which it was trained.

The GAN model includes two neural networks: a generator and a discriminator. The first network is intended for generating some quantities, such as particle characteristics. The second network identifies the differences between the values obtained by the generator and the real values.

The discriminator tries to distinguish the real values from those created by the generator, thus training it. The generator gets better at producing data with each training iteration, which in turn trains the discriminator [1].

While the GAN method has been successful for diverse applications (for example, generating photos and videos that are indistinguishable from real ones [2, 3]), it has certain drawbacks associated with complications in the training process of the model.

The reason for these complications is the strong dependence on the parameters of the model, often causing the following issues:

instabilities during training,

discrepancies,

parameter variations,

retraining of models.

There are many approaches to solving these problems, for example, those outlined in [4].

In this paper, we used the approach proposed in [5], described in detail below in the following section.

Applying GANs in high energy physics and elementary particle physics comes with additional difficulties. The most crucial are the multiple strict constraints dictated by conservation laws. Consequently, not every generation output can be considered suitable.

The prediction accuracy is also important; otherwise, the relationships between the derived quantities may be violated, which is also unacceptable. Similar problems are described, for example, in [6].

Conservation laws can produce significant irregularities in the distributions of physical quantities (for example, angles, momenta, energies, etc.) characterizing the interaction of particles. An example is the distribution of the p_z momentum component of the final-state lepton (Fig. 1,*a*). Multiplicity is understood (in Fig. 1 and below) as the number of events in the bin normalized by the total number of events, i.e., a dimensionless quantity. As evident from Fig. 1,*a*, the distribution has a sharp edge associated with the laws of conservation of energy–momentum: energy (or momentum) in the final state cannot exceed the level of energy (or momentum) in the initial state. The existence of such an irregularity negatively affects the training of GAN, as discussed in [6].

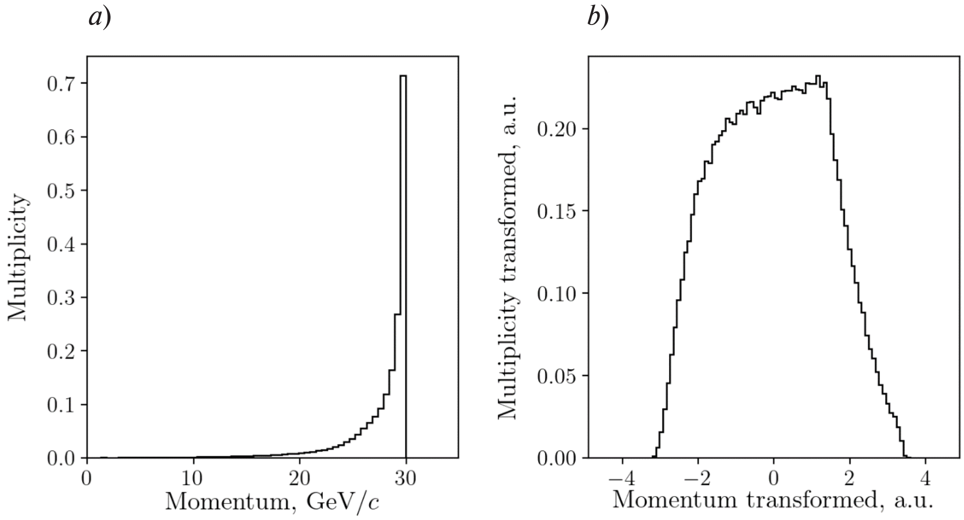

Fig. 1. Distributions of momentum component p_z of final-state lepton (*a*) and transformed quantity $T(p_z)$ (*b*)
Initial electron energy E_0 = 30 GeV

It was proposed in [7] that the problems associated with irregularities in the distributions of quantities with respect to specific physical parameters can be solved by generating transformed 'twins' of the quantities rather than the quantities themselves, modified in such a way that the new distribution becomes smoother.

The following transformation is used in this study for the p_z momentum component of the final-state lepton [7]:

$$T(p_z) = \log[(E_0 - p_z)/(1\ \text{GeV}/c)].$$

As a result, a smoother distribution is obtained (see Fig. 1,*b*).

A similar transformation was applied for the total energy of the scattered lepton E_l:

$$T(E_l) = \log[(E_0 - E_l)/(1\ \text{GeV}/c)].$$

Methodology

Since this paper considers inclusive scattering of charged leptons (e^+, e^-, μ^+, μ^-) by protons, the scattered lepton is characterized by four-momentum in the lepton–proton center-of-mass frame:

$$p_l = (E_l, \mathbf{p}),$$

where **p** is the lepton's three-dimensional momentum vector, given by the components p_x, p_y, p_z; E_l is the total energy of the scattered lepton.

Additional parameters are the total energy E_0 of the incident lepton in the lepton–proton center of mass frame and the type of lepton (e^+ or e^- or μ^+ or μ^-). These parameters allow the GAN to predict the final state of various leptons at different initial energies considered.

The energy E_0 is defined as

$$E_0 \approx \sqrt{s_{IN}}/2,$$

where $\sqrt{s_{IN}}$ is the initial energy in the lepton–proton center of mass frame.

The initial energies E_0 = 10, 20, 30, 40, 50 GeV were considered for training.

The PYTHIA8 program was used to obtain the final states of leptons [8]. 100,000 events were generated at initial energies $\sqrt{s_{IN}}$ = 20, 40, 60, 80 and 100 GeV for each type of lepton: (e^+, e^-, μ^+, μ^-). The four-momenta of the final-state lepton were recorded in each event (referred to as the real values).

Using the quantities $T(p_z)$ and $T(E_l)$ (the transformed quantities) allows the generator to avoid predicting unphysical values, and the discriminator to distinguish the real data from the generated ones.

The following quantities are fed to the discriminator input to increase its accuracy:

$$p_z,\ E_l,\ p_T = \sqrt{p_x^2 + p_y^2},\ \varphi = \arctan\left({p_z}/{p_T} \right),\ \theta = \arctan\left({p_y}/{p_x} \right)$$

(these are referred to as additional quantities).

A 128-dimensional noise vector (a vector of values obtained from a Gaussian distribution with the mean equal to 0 and the variance equal to 1), energy E_0 and lepton type are fed to the generator input. The generator network consists of 4 hidden layers of 512 neurons each with a Leaky ReLU activation function and a dropout of 0.2 [9]. The output layer consists of 4 neurons with a linear activation function. The output is four main predicted quantities: p_x, p_y, $T(p_z)$ and $T(E_l)$. In addition to these, the model includes the prediction of additional quantities: p_z, E_l, p_T, φ, θ, obtained based on the predicted ones. The main and additional quantities are then fed to the discriminator input.

The discriminator network consists of 4 hidden layers with 512 neurons each, a Leaky ReLU activation function and a dropout of 0.2 [9]. A so-called dropout layer with a rate of 10% [10] is applied to each of the layers, randomly dropping 10% of the layer weights. This helps prevent overfitting in classification procedures [11]. Spectral normalization is also applied to each layer [12], allowing to achieve a 1-Lipschitz mapping for the discriminator [13]. The output layer consists of a single neuron with a linear activation function. The higher the value obtained, the more confident the discriminator is in identifying the given values as realistic.

The paper uses the type of generative-adversarial network with a least square loss function.

The following expressions are valid for the loss functions of the discriminator (L_D) and the generator (L_G) in such networks [5]:

$$L_D = \frac{1}{2} E_{\boldsymbol{x} \sim p_{data}(\boldsymbol{x})}[(D(\boldsymbol{x} \mid \boldsymbol{y}) - b)^2] + \frac{1}{2} E_{z \sim p(z)}[(D(G(\boldsymbol{z} \mid \boldsymbol{y})) - a)^2], \tag{1}$$

$$L_G = \frac{1}{2} E_{z \sim p(z)}[(D(G(\boldsymbol{z} \mid \boldsymbol{y})) - c)^2], \tag{2}$$

where $D(...)$ is the discriminator network; $G(...)$ is the generator network; $\boldsymbol{x}$ are the real data; $\mathbf{z}$ is the noise vector; $D(\boldsymbol{x})$ are the values obtained by the discriminator based on the real data; $D(G(\mathbf{z}))$ are the values found by the discriminator based on the data obtained by the generator; E is the expected value; a, b are the hyperparameters of this loss function, equal to 0 and 1, respectively [5].

GAN was trained for 400 epochs in our study. RMSProp was used for gradient descent optimization, with $\rho = 0.9$ [14], $1\cdot10^{-4}$ training steps for the generator and $5\cdot10^{-5}$ for the discriminator. Using different training steps contributes to better training convergence, as shown in [15].

Simulation results

Due to the large number of possible scattering configurations (different types of leptons and different initial energies E_0), only some configurations are given below to illustrate the operation of the GAN.

Fig. 2 shows the distributions of the momentum components for the muon μ^+ and the electron e^- in the final states, obtained by GAN and the PYTHIA8 program. It can be seen that the model generates quantities with virtually identical distributions, as evidenced by the χ^2 values in the graphs and the corresponding momenta (p-value) [17].

Fig. 3 shows the distributions of the p_z momentum components of final-state electrons at different energies, obtained by GAN and the PYTHIA8 program. Analyzing the results obtained, we can conclude that the model can predict the correct distributions both at the energies at which the network was trained (10, 20, 30, 40, 50 GeV), and at interpolated energies (15, 25, 35, 45 GeV). Notably, the model can also predict the p_z values at high energies E_0 (60, 70, 80, 90 GeV).

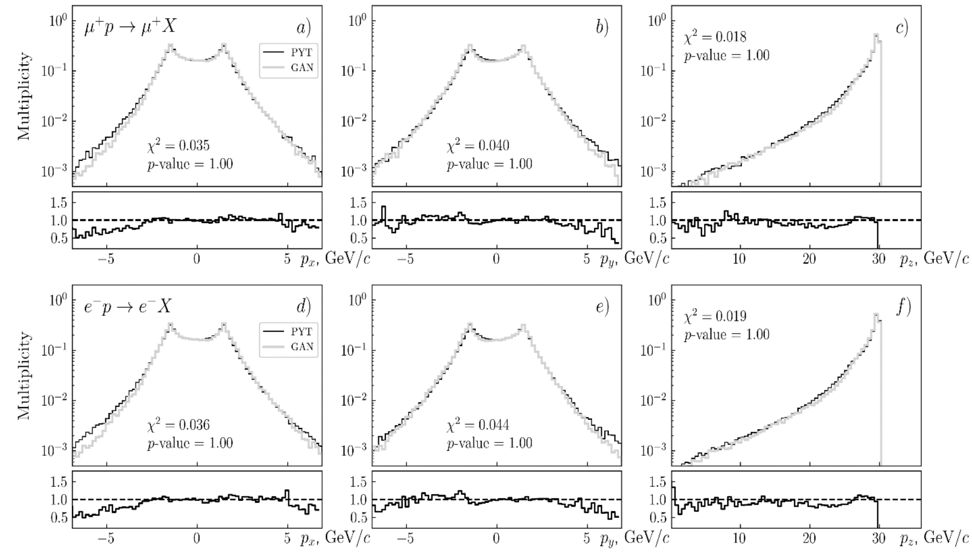

Fig. 2. Predicted distributions over momentum components p_x, p_y, p_z for the muon μ^+ (*a*, *b*, *c*) and electron e^- (*d*, *e*, *f*) at the same initial energy E_0 = 30 GeV, obtained using GAN (gray curves) and PYTHIA8 (black curves). The corresponding values of χ^2 and the graphs for the ratio of GAN to PYTHIA8 (GAN/PYT) predictions are given for each distribution

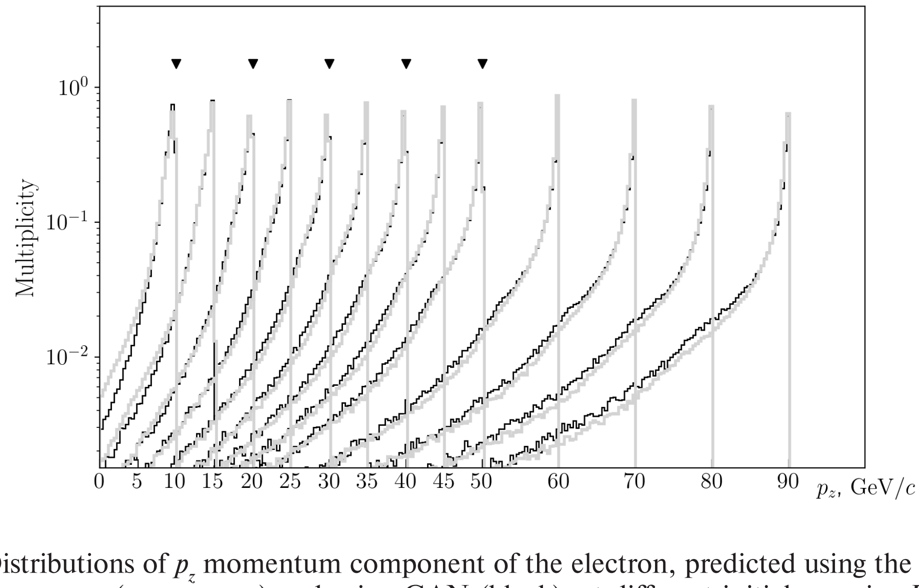

Fig. 3. Distributions of p_z momentum component of the electron, predicted using the PYTHIA8 program (gray curves) and using GAN (black), at different initial energies E_0
Triangles indicate the energies at which the model was trained

Aside from lepton momenta and energies, let us consider the quantities derived from them, used to characterize scattering. Such quantities include the squared momentum transfer $Q^2 = -q^2$ (q is the momentum of the virtual photon) and the Bjorken variable $x_{Bj} = Q^2/2Pq$ (P is the momentum of the incident proton).

Fig. 4 shows the joint distributions of Q^2 and x_{Bj} at energies E_0 = 10 and 40 GeV, obtained based on data from PYTHIA8 and GAN. Comparing the distributions in Fig. 4, *a* and *b* and those in Fig. 4, *c* and *d*, obtained by two approaches at two values of E_0 (10 and 40 GeV), we can see good agreement between the distributions obtained using PYTHIA8 and GCC. The χ^2 values calculated for all distribution bins are given as a quantitative assessment of this agreement.

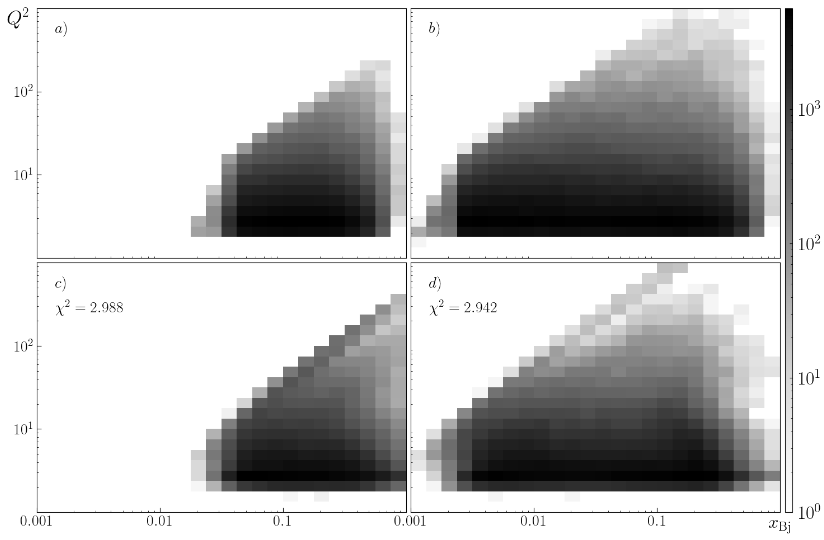

Fig. 4. Joint distribution of Q^2 and x_{Bj} for electrons at initial energies E_0 = 10 GeV (*a*, *c*) and 40 GeV (*b*, *d*), predicted using PYTHIA8 (*a*, *b*) and GAN (*c*, *d*)

The values of χ^2 are given to characterize the accuracy of GAN predictions for each value of E_0

Conclusion

The paper considers a generative-adversarial network (GAN) to generate the final state of leptons in inclusive deep inelastic lepton–proton scattering in the 20–100 GeV center-of-mass energy range.

We confirmed that the developed model can generate the distributions of various characteristics of different final-state leptons, including the quantities calculated based on the initially generated ones. The GAN can generate distributions not only at initial center-of-mass energies on which it was trained but also at interpolated energies (GeV): 15, 25, 35, 45.

In addition, we found that the model can generate the required distributions at extrapolated initial energies (GeV): 120, 140, 160 and 180.

In the future, there is a clear interest in considering semi-inclusive deep inelastic scattering, generating the characteristics of an additional particle, in particular a pion.

REFERENCES

1. **Goodfellow I., Pouget-Abadie J., Mirza M., et al.,** Generative Adversarial Networks, Commun. ACM. 63 (11) (2020) 139–144.

2. **Karras T., Laine S., Aila T.,** A style-based generator architecture for Generative Adversarial Networks, Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), Long Beach, USA, June 15–20 (2019) 4401–4410.

3. **Clark A., Donahue J., Simonyan K.,** Adversarial video generation on complex datasets, arXiv: 1907.06571v2, 2019. https://doi.org/10.48550/arXiv. 1907.06571.

4. **Gulrajani I., Ahmed F., Arjovsky M., et al.,** Improved training of Wasserstein GANs, arXiv:1704.00028v3, 2017. https: //doi.org/ 10.48550/arXiv. 1704.00028.

5. **Mao X., Li Q., Xie H., et al.,** On the effectiveness of least squares Generative Adversarial Networks, IEEE Trans. Pattern Anal. Mach. Intell. 41 (12) (2019) 2947–2960.

6. **Hashemi B., Amin N., Datta K., et al.,** LHC analysis-specific datasets with Generative Adversarial Networks. arXiv: 1901.05282v1, 2019. https: //doi.org/10.48550/arXiv.1901.05282.

7. **Alanazi Y., Sato N., Liu T., et al.,** Simulation of electron-proton scattering events by a Feature-Augmented and Transformed Generative Adversarial Network (FAT-GAN). arXiv: 2001.11103v2, 2019. https: //doi.org/10.48550/arXiv.2001. 11103.

8. **Sjöstrand T., Mrenna S., Skands P.,** A brief introduction to PYTHIA 8.1, Comput. Phys. Commun. 178 (11) (2008) 852–867.

9. **Sharma O.,** A new activation function for deep neural network, Proc. Int. Conf. Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), IEEE, Faridabad, India, Febr. 14–16 (2019) 84–86.

10. **Srivastava N., Hinton G., Krizhevsky A., et al.,** Dropout: A simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (2014) 1929–1958.

11. **Hawkins D. M.,** The problem of overfitting, J. Chem. Inf. Comput. Sci. 44 (1) (2003) 1–12.

12. **Miyato T., Kataoka T., Koyama M., Yoshida Y.,** Spectral normalization for Generative Adversarial Networks, arXiv: 1802.05957/v1, 2018. https://doi.org/10.48550/arXiv.1802.05957.

13. **Qin Y., Mitra N., Wonka C.,** How does Lipschitz regularization influence GAN training? Computer Vision – ECCV 2020, Springer Int. Publ. (2020) 310–326. https://doi.org/10.48550/arXiv.1811.09567

14. **Xu D., Zhang Sh., Zhang H., Mandic D. P.,** Convergence of the RMSProp deep learning method with penalty for nonconvex optimization, Neural Netw. 139 (July) (2021) 17–23.

15. **Heusel M., Ramsauer H., Unterthiner T., et al.,** GANs trained by a two time-scale update rule converge to a local Nash equilibrium, arXiv: 1706.08500v6, 2017. https://doi.org/10.48550/arXiv.1706.08500.

16. **McHugh M. L.,** The chi-square test of independence, Biochem. Med. 23 (2) (2013) 143–149.

СПИСОК ЛИТЕРАТУРЫ

1. **Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y.** Generative adversarial networks // Communications of the ACM. 2020. Vol. 63. No. 11. Pp. 139–144.

2. **Karras T., Laine S., Aila T.** A style-based generator architecture for generative adversarial networks // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, USA, June 15–20, 2019. Pp. 4401–4410.

3. **Clark A., Donahue J., Simonyan K.** Adversarial video generation on complex datasets. arXiv: 1907.06571v2, 2019. https://doi.org/10.48550/arXiv. 1907.06571.

4. **Gulrajani I., Ahmed F., Arjovsky M., Dumoulin V., Courville A.** Improved training of Wasserstein GANs. arXiv: 1704.00028v3, 2017. https: //doi.org/ 10.48550/arXiv.1704.00028.

5. **Mao X., Li Q., Xie H., Lau R. Y. K., Wang Zh., Smolley S. P.** On the effectiveness of least squares generative adversarial networks // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2019. Vol. 41. No. 12. Pp. 2947–2960.

6. **Hashemi B., Amin N., Datta K., Olivito D., Pierini N.** LHC analysis-specific datasets with Generative Adversarial Networks. arXiv: 1901.05282v1, 2019. https: //doi.org/10.48550/arXiv.1901.05282.

7. **Alanazi Y., Sato N., Liu T., et al.** Simulation of electron-proton scattering events by a Feature-Augmented and Transformed Generative Adversarial Network (FAT-GAN). arXiv: 2001.11103v2, 2019. https: //doi.org/10.48550/arXiv.2001. 11103.

8. **Sjöstrand T., Mrenna S., Skands P.** A brief introduction to PYTHIA 8.1 // Computer Physics Communications. 2008. Vol. 178. No. 11. Pp. 852–867.

9. **Sharma O.** A new activation function for deep neural network // Proceedings of the International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon). IEEE, Faridabad, India, February 14–16, 2019. Pp. 84–86.

10. **Srivastava N., Hinton G., Krizhevsky A., Slutskever I., Salakhutdinov R.** Dropout: A simple way to prevent neural networks from overfitting // The Journal of Machine Learning Research. 2014. Vol. 15. Pp. 1929–1958.

11. **Hawkins D. M.** The problem of overfitting // Journal of Chemical Information and Computer Sciences. 2003. Vol. 44. No. 1. Pp. 1–12.

12. **Miyato T., Kataoka T., Koyama M., Yoshida Y.** Spectral normalization for Generative Adversarial Networks. arXiv: 1802.05957/v1, 2018. https://doi.org/
10.48550/arXiv.1802.05957.

13. **Qin Y., Mitra N., Wonka C.** How does Lipschitz regularization influence GAN training? // Computer Vision – ECCV 2020. Springer International Publishing, 2020. Pp. 310–326. https://doi.org/10.48550/arXiv.1811.09567

14. **Xu D., Zhang Sh., Zhang H., Mandic D. P.** Convergence of the RMSProp deep learning method with penalty for nonconvex optimization // Neural Networks. 2021. Vol. 139. July. Pp. 17–23.

15. **Heusel M., Ramsauer H., Unterthiner T., Nessler B., Hochreiter S.** GANs trained by a two time-scale update rule converge to a local Nash equilibrium. arXiv: 1706.08500v6, 2017. https://doi.org/10.48550/arXiv.1706. 08500.

16. **McHugh M. L.** The chi-square test of independence // Biochemia Medica. 2013. Vol. 23. No. 2. Pp. 143–149.

THE AUTHORS

LOBANOV Andrey A.
Peter the Great St. Petersburg Polytechnic University
29 Politechnicheskaya St., St. Petersburg, 195251, Russia
lobanov2.aa@edu.spbstu.ru
ORCID: 0000-0002-8910-4775

BERDNIKOV Yaroslav A.
Peter the Great St. Petersburg Polytechnic University
29 Politechnicheskaya St., St. Petersburg, 195251, Russia
berdnikov@spbstu.ru
ORCID: 0000-0003-0309-5917

СВЕДЕНИЯ ОБ АВТОРАХ

ЛОБАНОВ Андрей Александрович – *студент Физико-механического института Санкт-Петербургского политехнического университета Петра Великого.*
195251, Россия, г. Санкт-Петербург, Политехническая ул., 29
lobanov2.aa@edu.spbstu.ru
ORCID: 0000-0002-8910-4775

БЕРДНИКОВ Ярослав Александрович – *доктор физико-математических наук, профессор Высшей школы фундаментальных физических исследований Санкт-Петербургского политехнического университета Петра Великого.*
195251, Россия, г. Санкт-Петербург, Политехническая ул., 29
berdnikov@spbstu.ru
ORCID: 0000-0003-0309-5917

Received 20.07.2023. Approved after reviewing 31.07.2023. Accepted 31.07.2023.
Статья поступила в редакцию 20.07.2023. Одобрена после рецензирования 31.07.2023. Принята 31.07.2023.