# A MODERN MACHINE LEARNING APPROACH FOR $B$-MESON DECAY GENERATIVE MODELING*

Pawel Kopciewicz, Michal Kacprzak, Tomasz Szumlak

AGH University of Science and Technology, al. Mickiewicza 30, Kraków, Poland

*Received 1 October 2022, accepted 2 January 2023,*
*published online 15 February 2023*

Recently, generative Machine Learning methods have gained a number of applications in physics analysis and detector science as they prove to be an excellent tool for data modeling. Generative Adversarial Networks (GANs), invented in 2014, were a breakthrough in computer vision and image synthesis, *e.g.* being able to generate realistic human faces. Although these networks show a growing presence in High-Energy Physics, their applications in this field are still quite scarce. The document investigates the potential use of GANs in Monte Carlo (MC) data generation in physics analysis, on the example of $B^0 \to D_s^- \pi^+$ decay, simulated under the LHCb spectrometer geometry, and investigates whether they could be potentially used as an alternative method to the existing MC generators. Although GAN can quickly learn the shape of the distribution and recreate some correlations between physics parameters, its disadvantage is the lack of precision. The paper gives an overview of GAN capabilities as well as the optimized configuration of the model.

DOI:10.5506/APhysPolBSupp.16.3-A5

## 1. Introduction

The Monte Carlo (MC) data simulation and data multiplication are common problems in physics data analysis. Laborious MC studies are often necessary to grasp the relationship between similar decays after physics selection and to carefully model the mass spectrum from the generated data, such that a similar model could be used for the real physics data. The paper investigates the potential use of the modern Machine Learning (ML) method to recreate the mass spectrum using Generative Adversarial Networks (GAN). The signal decay was chosen to be $B^0 \to D_s^- \pi^+$ and was studied assuming the LHCb [1] experiment's spectrometer geometry [2].

* Presented at the XIV International Conference on *Beauty, Charm and Hyperon Hadrons*, Kraków, Poland, 5–10 June, 2022.

GANs [3] are lately brought up as the most effective generative models, which found a variety of applications, from computer vision [4], image synthesis [5], semantic segmentation, and image-to-image translation to natural language processing. GAN is a complex system built of two components, one generative and one discriminative, which are deep neural networks (for a detailed explanation of neural networks and deep processing, one could refer to [6]). The generator tries to fool the discriminator and generate objects that are possibly close to the ones coming from the real dataset, such that the discriminator cannot recognize which one is real or fake. At the same time, the discriminator tries to correctly classify the objects and to tell whether they are real or fake. A scheme of a simple GAN model and its training is shown in Fig. 1.

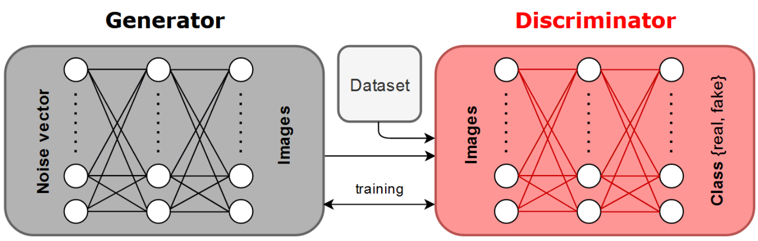


Fig. 1. The generic scheme of GAN.

The training of the model can be expressed as follows, the generator $G$ and the discriminator $D$ follow a mini–max strategy, and according to the computed loss, their weights (latent space parameters) are updated by making steps in the direction toward the highest descent of the gradient of the loss. In the Goodfellow model [3], the loss function of the generator was a binary cross-entropy (1)

$$L_{\text{gen}} = \min[\log(D(x)) + \log(1 - D(G(n)))]\,, \tag{1}$$

where $x$ is a real data sample, and $G(n)$ is a fake data sample generated from the noise array $n$. The discriminator $D$ does not update the weights, while $G$ computes its updated weights. Because both $G$ and $D$ aim to minimize their losses, the mini–max strategy could be expressed, according to Goodfellow, as

$$\min_G \max_D V(D,G) = E\left(X_{p_{\text{data}}(x)}[\log(D(x))]\right) + E\left(Y_{p_y(y)}[\log(1-D(G(y)))]\right)\,, \tag{2}$$

where $D(X)$ is the probability estimation of a real sample being categorized correctly, $D(G(y))$ is a probability estimation that a fake sample is judged as false, $EX$ and $EY$ are expected values when a whole dataset is taken into account, and finally, $V(D,G)$ is the adversarial loss.

In 2017, Arjovsky [7] replaced Goodfellow's adversarial loss with an alternative, which nowadays is known as Wasserstein loss [8], as it refers to the Wasserstein metric. The Wasserstein metric, or Wasserstein distance, is in this case the distance between the real and generated datasets, computed as a multiplication of the number of moves required to change the shape of the distribution times the Euclidean distance between both distributions. The Wasserstein GAN (WGAN) proved to be much more efficient in image generation, and the study presented in this document is, in fact, based on this variant of the model.

The main problem of WGAN in physics data generation is their lack of mathematical precision. While WGANs are powerful in image generation, there is no big difference for the image if a pixel or group of pixels change their color indices or the ratio between colors intensity. Such fluctuations cannot be happening in complex mathematical modeling. This issue can be addressed by reducing the dimensionality of the input features. The studies presented in this paper use only six real values of the kinematical variables, more precisely, the decay products momentum of $D_s^-$ and $\pi^+$ in $B^0 \to D_s^- \pi^+$. This way, the number of degrees of freedom is significantly lower than in the image processing WGAN proved to be very effective in. As explored in the studies, some more complex mathematical formulas might be troublesome to reflect due to the stochastic gradient descent limitations.

## 2. The model analysis

A data sample of $10^5$ events of $B^0 \to D_s^- \pi^+$ was generated using RapidSim [10], an engine with implemented the LHCb detector geometry. PyTorch was chosen as an environment for the model (a Python library for deep learning), introducing WGAN implemented earlier by Lindernoren [11], and trained on the dataset from RapidSim. GAN started to generate the physics events, and because the whole model is very complex, it was then adjusted for the optimal hyper-parameter configuration in many steps by a grid search. The setup found as optimal for this type of physics event generation was as follows: $G$ architecture $10\times300\times600\times6$, $D$ architecture $6\times600\times300\times2$, the learning rate $1.6\times10^{-4}$, batch size 500, the maximal number of epochs 500, optimizer ADAM (an improved stochastic gradient descent) with $\beta_1 = 0.5$ (the exponent's index of decay rate in ADAM's gradient weighting), and the 6-dimensional latent noise input array. Such a latent input space setup was chosen to push the network to learn only a linear transformation between the uncorrelated noise base to the base of six real values of the momentum.

The training of the model is visualized in Fig. 2, with $G$ and $D$ loss and the prediction accuracy of $D$, which approaches 0.5 at the end of the training. The convergence means that $D$ and $G$ train at the same pace, but the training has to be stopped before the losses reach 0.5, as then the generator can see the discriminator behavior as random, and the feedback it gives gets less valuable. $G$ and $D$ loss approach 0.75 as the mini–max strategy they follow reaches the point of equilibrium. This perfect training of the model is usually hard to achieve in computer vision and was possible mostly due to the low number of degrees of freedom.

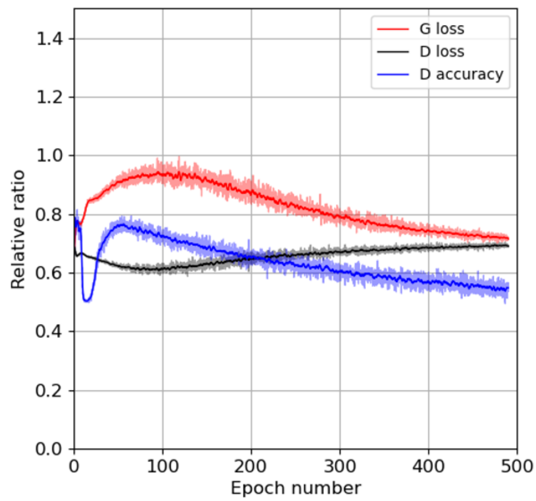


Fig. 2. The generator and discriminator $G$ and $D$ loss, and the prediction accuracy of the discriminator $D$.

After the training, WGAN was able to generate the physics parameters with distribution shapes corresponding to those from the training dataset. Figure 3 presents one of the generated real values, the $X$ momentum of $D_s^-$, and the comparison to the training dataset from RapidSim. The distribution is reflected quite nicely but the very peak, where the number of events is slightly underestimated. When the momentum vector of that product is taken in a whole in the form of three real values, it can be used to calculate $D_s^-$ total energy (provided the mass of $D_s^-$ is known from PDG $m_{D_s^-} = 1969 \pm 1.4$ MeV). When this expression is computed using the generated kinematical variables, the resulting distribution of energy is consistent with that obtained from the training dataset as well, which is shown in Fig. 4.

Unlike the total energy of the product, a mass of the $B_0$ is given by a much more complex function (a function of six real values), and unfortunately, WGAN is not able to reconstruct the correlations deep enough to make the mass distribution to be correctly reconstructed, or at least to come close to the uncertainties that can be achieved using standard MC generators. The idea proposed by Szumlak is to further modify the loss function (pursuing Arjovsky's idea) and to add to the loss function an additional cost that would reflect the nature of the mathematical expressions used in physics analysis, which operate on the squares of the variables (the momenta are always squared when computing the energy and mass).

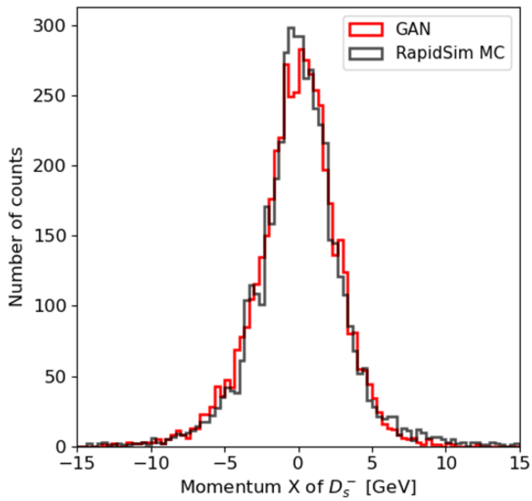


Fig. 3. The reconstructed $X$ momentum of $D_s^-$ using the Wasserstein GAN.

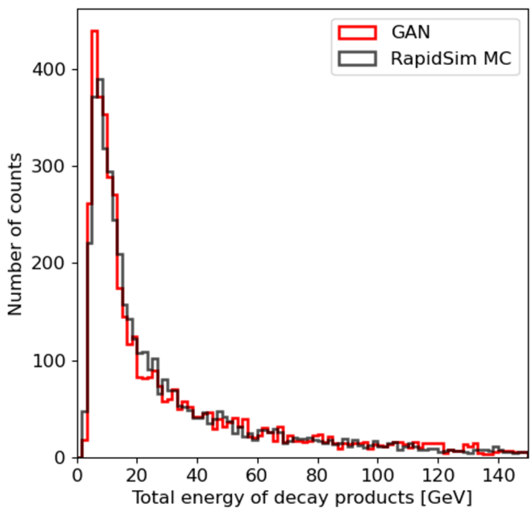


Fig. 4. The energy of $D_s^-$ computed using kinematical variables generated by WGAN.

## 3. Conclusions

The paper presents preliminary studies of using modern generative systems in the MC data generation problem. The most advanced method in the field, Generative Adversarial Network, was applied to the example of $B^0 \to D_s^- \pi^+$ decay and was optimized for the considered problem, with the optimal configuration given in this document. The Wasserstein GAN can recreate the physics events of the decay to some precision. Further research on this matter is very promising, in particular after the success, which generative models achieved lately in Computer Science.

We acknowledge support from the National Science Centre, Poland (NCN) (UMO-2020/37/N/ST2/04008).

REFERENCES

[1] LHCb Collaboration (F. Filthaut), «The LHCb Physics Programme», in: «Proceedings of 34$^{\text{th}}$ Rencontres de Moriond on QCD and High Energy Hadronic Interactions», Les Arcs, France, 1999, pp. 409–414.

[2] LHCb Collaboration, «Framework TDR for the LHCb Upgrade: Technical Design Report», CERN-LHCC-2012-007, 2012

[3] I. Goodfellow *et al.*, «Generative adversarial nets», in: «Advances in Neural Information Processing Systems», 2014, pp. 2672–2680.

[4] Z. Wang *et al.*, *ACM Comput. Surv.* **54**, 1 (2021).

[5] M. Gupta *et al.*, «Chapter 1 — Super-resolution-based GAN for image processing: Recent advances and future trends», in: «Generative Adversarial Networks for Image-to-Image Translation», 2021, pp. 1–15.

[6] I. Goodfellow *et al.*, «Deep Learning», *MIT Press*, 2016.

[7] M. Arjovsky *et al.*, arXiv:1701.07875 [stat.ML].

[8] L. Rüschendorf, *Probab. Theory Related Fields* **70**, 117 (1985).

[9] A. Aggarwal, M. Mittal, G. Battineni, *Int. J. Inform. Manage.* **1**, 100004 (2021).

[10] G. Cowan *et al.*, «RapidSim v1.4.2, documentation and software», available at https://zenodo.org/record/1068329#.YtLfrYTP1PY

[11] «PyTorch implementation of GANs», available at https://github.com/eriklindernoren/PyTorch-GAN

[12] Particle Data Group (R.L. Workmanet *et al.*), *Prog. Theor. Exp. Phys.* **2022**, 083C01 (2022).