Journal of Instrumentation

SISSA

Fast simulation of muons produced at the SHiP experiment using Generative Adversarial Networks

To cite this article: C. Ahdida *et al* 2019 *JINST* **14** P11028

View the article online for updates and enhancements.

Jinst

Published by IOP Publishing for Sissa Medialab

Received: *September 13, 2019*
Accepted: *November 4, 2019*
Published: *November 27, 2019*

Fast simulation of muons produced at the SHiP experiment using Generative Adversarial Networks

SHiP
Search for Hidden Particles

The SHiP collaboration

C. Ahdida,[44] R. Albanese,[14,d,h] A. Alexandrov,[14] A. Anokhina,[39] S. Aoki,[18] G. Arduini,[44] E. Atkin,[38] N. Azorskiy,[29] J.J. Back,[54] A. Bagulya,[32] F. Baaltasar Dos Santos,[44] A. Baranov,[40] F. Bardou,[44] G.J. Barker,[54] M. Battistin,[44] J. Bauche,[44] A. Bay,[46] V. Bayliss,[51] G. Bencivenni,[15] A.Y. Berdnikov,[37] Y.A. Berdnikov,[37] I. Berezkina,[32] M. Bertani,[15] C. Betancourt,[47] I. Bezshyiko,[47] O. Bezshyyko,[55] D. Bick,[8] S. Bieschke,[8] A. Blanco,[28] J. Boehm,[51] M. Bogomilov,[1] K. Bondarenko,[27,55] W.M. Bonivento,[13] J. Borburgh,[44] A. Boyarsky,[27,55] R. Brenner,[43] D. Breton,[4] R. Brundler,[47] M. Bruschi,[12] V. Büscher,[10] A. Buonaura,[47] S. Buontempo,[14] S. Cadeddu,[13] A. Calcaterra,[15] M. Calviani,[44] M. Campanelli,[53] M. Casolino,[44] N. Charitonidis,[44] P. Chau,[10] J. Chauveau,[5] A. Chepurnov,[39] M. Chernyavskiy,[32] K.-Y. Choi,[26] A. Chumakov,[2] P. Ciambrone,[15] L. Congedo,[11,a] K. Cornelis,[44] M. Cristinziani,[7] A. Crupano,[14,d] G.M. Dallavalle,[12] A. Datwyler,[47] N. D'Ambrosio,[16] G. D'Appollonio,[13,c] J. De Carvalho Saraiva,[28] G. De Lellis,[14,34,44,d] M. de Magistris,[14,d] A. De Roeck,[44] M. De Serio,[11,a] D. De Simone,[14,d] L. Dedenko,[39] P. Dergachev,[34] A. Di Crescenzo,[14,d] N. Di Marco,[16] C. Dib,[2] H. Dijkstra,[44] P. Dipinto,[11,a] V. Dmitrenko,[38] S. Dmitrievskiy,[29] L.A. Dougherty,[44] A. Dolmatov,[30] D. Domenici,[15] S. Donskov,[35] V. Drohan,[55] A. Dubreuil,[45] M. Ehlert,[6] T. Enik,[29] A. Etenko,[33,38] F. Fabbri,[12] L. Fabbri,[12,b] A. Fabich,[44] O. Fedin,[36] F. Fedotovs,[52] G. Felici,[15] M. Ferro-Luzzi,[44] K. Filippov,[38] R.A. Fini,[11] P. Fonte,[28] C. Franco,[28] M. Fraser,[44] R. Fresa,[14,i] R. Froeschl,[44] T. Fukuda,[19] G. Galati,[14,d] J. Gall,[44] L. Gatignon,[44] G. Gavrilov,[38] V. Gentile,[14,d] S. Gerlach,[6] B. Goddard,[44] L. Golinka-Bezshyyko,[55] A. Golovatiuk,[14,d] D. Golubkov,[30] A. Golutvin,[52] P. Gorbounov,[44] D. Gorbunov,[31] S. Gorbunov,[32] V. Gorkavenko,[55] Y. Gornushkin,[29] M. Gorshenkov,[34] V. Grachev,[38] A.L. Grandchamp,[46] G. Granich,[32] E. Graverini,[46] J.-L. Grenard,[44] D. Grenier,[44] V. Grichine,[32] N. Gruzinskii,[36] A. M. Guler,[48] Yu. Guz,[35] G.J. Haefeli,[46] C. Hagner,[8] H. Hakobyan,[2] I.W. Harris,[46] E. van Herwijnen,[44] C. Hessler,[44] A. Hollnagel,[10] B. Hosseini,[52] M. Hushchyn,[40] G. Iaselli,[11,a] A. Iuliano,[14,d] V. Ivantchenko,[32]

https://doi.org/10.1088/1748-0221/14/11/P11028

R. Jacobsson,[44] D. Joković,[41] M. Jonker,[44] I. Kadenko,[55] V. Kain,[44] B. Kaiser,[8]
C. Kamiscioglu,[49] K. Kershaw,[44] M. Khabibullin,[31] E. Khalikov,[39] G. Khaustov,[35]
G. Khoriauli,[10] A. Khotyantsev,[31] S.H. Kim,[22] Y.G. Kim,[23] V. Kim,[36,37] N. Kitagawa,[19]
J.-W. Ko,[22] K. Kodama,[17] A. Kolesnikov,[29] D.I. Kolev,[1] V. Kolosov,[35] M. Komatsu,[19]
N. Kondrateva,[32] A. Kono,[21] N. Konovalova,[32,34] S. Kormannshaus,[10] I. Korol,[6] I. Korol'ko,[30]
A. Korzenev,[45] V. Kostyukhin,[7] E. Koukovini Platia,[44] S. Kovalenko,[2] I. Krasilnikova,[34]
Y. Kudenko,[31,38,g] E. Kurbatov,[40] P. Kurbatov,[34] V. Kurochka,[31] E. Kuznetsova,[36]
H.M. Lacker,[6] M. Lamont,[44] G. Lanfranchi,[15] O. Lantwin,[52] A. Lauria,[14,d] K.S. Lee,[25]
K.Y. Lee,[22] J.-M. Lévy,[5] V.P. Loschiavo,[14,h] L. Lopes,[28] E. Lopez Sola,[44] V. Lyubovitskij,[2]
J. Maalmi,[4] A. Magnan,[52] V. Maleev,[36] A. Malinin,[33] Y. Manabe,[19] A.K. Managadze,[39]
M. Manfredi,[44] S. Marsh,[44] A.M. Marshall,[50] A. Mefodev,[31] P. Mermod,[45] A. Miano,[14,d]
S. Mikado,[20] Yu. Mikhaylov,[35] D.A. Milstead,[42] O. Mineev,[31] A. Montanari,[12] M.C. Montesi,[14,d]
K. Morishima,[19] S. Movchan,[29] Y. Muttoni,[44] N. Naganawa,[19] M. Nakamura,[19] T. Nakano,[19]
S. Nasybulin,[36] P. Ninin,[44] A. Nishio,[19] A. Novikov,[38] B. Obinyakov,[33] S. Ogawa,[21]
N. Okateva,[32,34] B. Opitz,[8] J. Osborne,[44] M. Ovchynnikov,[27,55] N. Owtscharenko,[7]
P.H. Owen,[47] P. Pacholek,[44] A. Paoloni,[15] B.D. Park,[22] S.K. Park,[25] A. Pastore,[11] M. Patel,[52]
D. Pereyma,[30] A. Perillo-Marcone,[44] G.L. Petkov,[1] K. Petridis,[50] A. Petrov,[33] D. Podgrudkov,[39]
V. Poliakov,[35] N. Polukhina,[32,34,38] J. Prieto Prieto,[44] M. Prokudin,[30] A. Prota,[14,d]
A. Quercia,[14,d] A. Rademakers,[44] A. Rakai,[44] F. Ratnikov,[40] T. Rawlings,[51] F. Redi,[46]
S. Ricciardi,[51] M. Rinaldesi,[44] Volodymyr Rodin,[55] Viktor Rodin,[55] P. Robbe,[4]
A.B. Rodrigues Cavalcante,[46] T. Roganova,[39] H. Rokujo,[19] G. Rosa,[14,d] T. Rovelli,[12,b]
O. Ruchayskiy,[3] T. Ruf,[44] V. Samoylenko,[35] V. Samsonov,[38] F. Sanchez Galan,[44]
P. Santos Diaz,[44] A. Sanz Ull,[44] A. Saputi,[15] O. Sato,[19] E.S. Savchenko,[34] J.S. Schliwinski,[6]
W. Schmidt-Parzefall,[8] N. Serra,[47] S. Sgobba,[44] O. Shadura,[55] A. Shakin,[34]
M. Shaposhnikov,[46] P. Shatalov,[30] T. Shchedrina,[32,34] L. Shchutska,[46] V. Shevchenko,[33]
H. Shibuya,[21] L. Shihora,[6] S. Shirobokov,[52] A. Shustov,[38] S.B. Silverstein,[42] S. Simone,[11,a]
R. Simoniello,[10] M. Skorokhvatov,[38,33] S. Smirnov,[38] J.Y. Sohn,[22] A. Sokolenko,[55]
E. Solodko,[44] N. Starkov,[32,33] L. Stoel,[44] B. Storaci,[47] M.E. Stramaglia,[46] D. Sukhonos,[44]
Y. Suzuki,[19] S. Takahashi,[18] J.L. Tastet,[3] P. Teterin,[38] S. Than Naing,[32] I. Timiryasov,[46]
V. Tioukov,[14] D. Tommasini,[44] M. Torii,[19] N. Tosi,[12] D. Treille,[44] R. Tsenov,[1,29] S. Ulin,[38]
A. Ustyuzhanin,[40] Z. Uteshev,[38] G. Vankova-Kirilova,[1] F. Vannucci,[5] P. Venkova,[6] V. Venturi,[44]
S. Vilchinski,[55] M. Villa,[12,b] Heinz Vincke,[44] Helmut Vincke,[44] C. Visone,[14,d] K. Vlasik,[38]
A. Volkov,[32,33] R. Voronkov,[32] S. van Waasen,[9] R. Wanke,[10] P. Wertelaers,[44] J.-K. Woo,[24]
M. Wurm,[10] S. Xella,[3] D. Yilmaz,[49] A.U. Yilmazer,[49] C.S. Yoon,[22] P. Zarubin,[29] I. Zarubina[29]
and Yu. Zaytsev[30]

[1]*Faculty of Physics, Sofia University, Sofia, Bulgaria*

[2]*Universidad Técnica Federico Santa María and Centro Científico Tecnológico de Valparaíso, Valparaíso, Chile*

[3]*Niels Bohr Institute, University of Copenhagen, Copenhagen, Denmark*

[4]*LAL, Univ. Paris-Sud, CNRS/IN2P3, Université Paris-Saclay, Orsay, France*

[5]*LPNHE, IN2P3/CNRS, Sorbonne Université, Université Paris Diderot,F-75252 Paris, France*

[6]*Humboldt-Universität zu Berlin, Berlin, Germany*

[7]*Physikalisches Institut, Universität Bonn, Bonn, Germany*

[8]*Universität Hamburg, Hamburg, Germany*
[9]*Forschungszentrum Jülich GmbH (KFA), Jülich, Germany*
[10]*Institut für Physik and PRISMA Cluster of Excellence, Johannes Gutenberg Universität Mainz, Mainz, Germany*
[11]*Sezione INFN di Bari, Bari, Italy*
[12]*Sezione INFN di Bologna, Bologna, Italy*
[13]*Sezione INFN di Cagliari, Cagliari, Italy*
[14]*Sezione INFN di Napoli, Napoli, Italy*
[15]*Laboratori Nazionali dell'INFN di Frascati, Frascati, Italy*
[16]*Laboratori Nazionali dell'INFN di Gran Sasso, L'Aquila, Italy*
[17]*Aichi University of Education, Kariya, Japan*
[18]*Kobe University, Kobe, Japan*
[19]*Nagoya University, Nagoya, Japan*
[20]*College of Industrial Technology, Nihon University, Narashino, Japan*
[21]*Toho University, Funabashi, Chiba, Japan*
[22]*Physics Education Department & RINS, Gyeongsang National University, Jinju, Korea*
[23]*Gwangju National University of Education,*[e] *Gwangju, Korea*
[24]*Jeju National University,*[e] *Jeju, Korea*
[25]*Korea University, Seoul, Korea*
[26]*Sungkyunkwan University,*[e] *Suwon-si, Gyeong Gi-do, Korea*
[27]*University of Leiden, Leiden, The Netherlands*
[28]*LIP, Laboratory of Instrumentation and Experimental Particle Physics, Portugal*
[29]*Joint Institute for Nuclear Research (JINR), Dubna, Russia*
[30]*Institute of Theoretical and Experimental Physics (ITEP) NRC 'Kurchatov Institute', Moscow, Russia*
[31]*Institute for Nuclear Research of the Russian Academy of Sciences (INR RAS), Moscow, Russia*
[32]*P.N. Lebedev Physical Institute (LPI), Moscow, Russia*
[33]*National Research Centre 'Kurchatov Institute', Moscow, Russia*
[34]*National University of Science and Technology "MISiS", Moscow, Russia*
[35]*Institute for High Energy Physics (IHEP) NRC 'Kurchatov Institute', Protvino, Russia*
[36]*Petersburg Nuclear Physics Institute (PNPI) NRC 'Kurchatov Institute', Gatchina, Russia*
[37]*St. Petersburg Polytechnic University (SPbPU),*[f] *St. Petersburg, Russia*
[38]*National Research Nuclear University (MEPhI), Moscow, Russia*
[39]*Skobeltsyn Institute of Nuclear Physics of Moscow State University (SINP MSU), Moscow, Russia*
[40]*Yandex School of Data Analysis, Moscow, Russia*
[41]*Institute of Physics, University of Belgrade, Serbia*
[42]*Stockholm University, Stockholm, Sweden*
[43]*Uppsala University, Uppsala, Sweden*
[44]*European Organization for Nuclear Research (CERN), Geneva, Switzerland*
[45]*University of Geneva, Geneva, Switzerland*
[46]*École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland*
[47]*Physik-Institut, Universität Zürich, Zürich, Switzerland*
[48]*Middle East Technical University (METU), Ankara, Turkey*
[49]*Ankara University, Ankara, Turkey*
[50]*H.H. Wills Physics Laboratory, University of Bristol, Bristol, U.K.*

[51] *STFC Rutherford Appleton Laboratory, Didcot, U.K.*
[52] *Imperial College London, London, U.K.*
[53] *University College London, London, U.K.*
[54] *University of Warwick, Warwick, U.K.*
[55] *Taras Shevchenko National University of Kyiv, Kyiv, Ukraine*
[a] *Università di Bari, Bari, Italy*
[b] *Università di Bologna, Bologna, Italy*
[c] *Università di Cagliari, Cagliari, Italy*
[d] *Università di Napoli "Federico II", Napoli, Italy*
[e] *Associated to Gyeongsang National University, Jinju, Korea*
[f] *Associated to Petersburg Nuclear Physics Institute (PNPI), Gatchina, Russia*
[g] *Also at Moscow Institute of Physics and Technology (MIPT), Moscow Region, Russia*
[h] *Consorzio CREATE, Napoli, Italy*
[i] *Università della Basilicata, Potenza, Italy*

E-mail: alex.marshall@cern.ch

Abstract: This paper presents a fast approach to simulating muons produced in interactions of the SPS proton beams with the target of the SHiP experiment. The SHiP experiment will be able to search for new long-lived particles produced in a 400 GeV/c SPS proton beam dump and which travel distances between fifty metres and tens of kilometers. The SHiP detector needs to operate under ultra-low background conditions and requires large simulated samples of muon induced background processes. Through the use of Generative Adversarial Networks it is possible to emulate the simulation of the interaction of 400 GeV/c proton beams with the SHiP target, an otherwise computationally intensive process. For the simulation requirements of the SHiP experiment, generative networks are capable of approximating the full simulation of the dense fixed target, offering a speed increase by a factor of $O(10^6)$. To evaluate the performance of such an approach, comparisons of the distributions of reconstructed muon momenta in SHiP's spectrometer between samples using the full simulation and samples produced through generative models are presented. The methods discussed in this paper can be generalised and applied to modelling any non-discrete multi-dimensional distribution.

Keywords: Detector modelling and simulations I (interaction of radiation with matter, interaction of photons with matter, interaction of hadrons with matter, etc); Simulation methods and programs

ArXiv ePrint: 1909.04451

Contents

1 Introduction

Generative networks are a class of machine learning algorithms designed to generate samples according to a multidimensional function, given a randomly distributed input sample. Generative networks have been studied in the machine learning community primarily for the purpose of image generation. Each image in a training set is made up of a multitude of pixels, corresponding to a data point in a high dimensional space. Within this space, underlying features of the set of images are encoded through dependencies between pixels. Generative networks attempt to model the characteristics that define a specific set of training images. These models can then be used to generate images that are faithful emulations of the original training set. Generative networks have been successfully employed for a variety of applications such as: generating high quality images that obey fundamental features of training set images; the generation of images from descriptive text [1]; modelling image captions [2]; producing photo realistic super resolution images [3]; and generating high resolution images from semantic mapping [4, 5].

Searches for physics beyond the Standard Model often involve looking for rare signatures and must therefore be able to suppress background processes which can be many orders of magnitude more abundant than the signal. In order to optimise the design of the detectors, develop reconstruction algorithms and understand the efficiency of the selection criteria, large samples of simulated background events are required. Dedicated software packages such as GEANT4 [6] model the transport of particles through the material and the detector response. In many cases, the CPU requirements to simulate these interactions with matter prohibit the production of large numbers of background events due to the computationally expensive procedure. Therefore, the computing demands of the simulation of high energy physics experiments are increasing exponentially [7]. Recent algorithmic improvements that take advantage of high performance computing resources aim at reducing simulation time, resulting in an order of magnitude increase in speed [8]. This improvement is not sufficient to meet the demands of future particle physics experiments, such as those at the High Luminosity LHC, for large simulation samples [7]. Generative neural networks offer an alternative approach to simulation by modelling non-analytical functions in a computationally efficient way [9].

The use of generative networks for particle physics originally focused on image based generation. Examples of their application include: the development of Location Aware Generative Adversarial Networks for the production of images of jets [10]; the simulation of reconstructed cosmic ray induced air showers [11], and of showers in electromagnetic calorimeters [12]; the fast simulation of Cherenkov detectors [13]. More recently, generative approaches have also been adopted to simulate the kinematics of final state particles emerging from physical exclusive two-to-two processes, such as Z or top-quark production at the LHC [14, 15]. In ref. [16], generative networks were used to simulate the detector reconstruction of the $Z \to \mu^+\mu^-$ process at the LHC.

This paper describes the use of generative networks to emulate the kinematics of muons produced through the interactions of high energy protons with the dense target designed for the Search for Hidden Particles (SHiP) experiment [17]. This approach offers a gain of multiple orders of magnitude in the computational efficiency of such processes. In contrast to the aforementioned use of generative networks to approximate a single exclusive process, this work employs four different generative networks to model the kinematics of muons originating from a multitude of processes, including muons from secondary interactions or particle showers in SHiP's target. Large samples of generative based muons can then be passed through the GEANT4 based simulation of the rest of the SHiP experiment, offering a precise modelling of its detector response and reconstruction.

This paper is organised as follows: sections 2 and 3 describe the SHiP detector and its simulation framework respectively; section 4 discusses the generative models used in this analysis, while section 5 details how these networks are trained and optimised for the SHiP experiment. Section 6 and 7 then present the performance of the generative models in simulating muons produced through interactions of high energy protons with the SHiP target, compared to the Pythia8 [18] and GEANT4 frameworks. Finally, section 8 discusses the computational time required to produce muons through generative networks.

2 The SHiP experiment

The Search for Hidden Particles experiment (SHiP) is a proposed experiment that will operate at the the prospective general purpose fixed target facility at the CERN Super Proton Synchrotron (SPS) accelerator. The SHiP experiment aims to search for long-lived exotic particles with masses between a few hundred MeV$/c^2$ and a few GeV$/c^2$. These particles are expected to be produced in the decays of heavy hadrons. The facility is therefore designed to maximise the production rate and detection

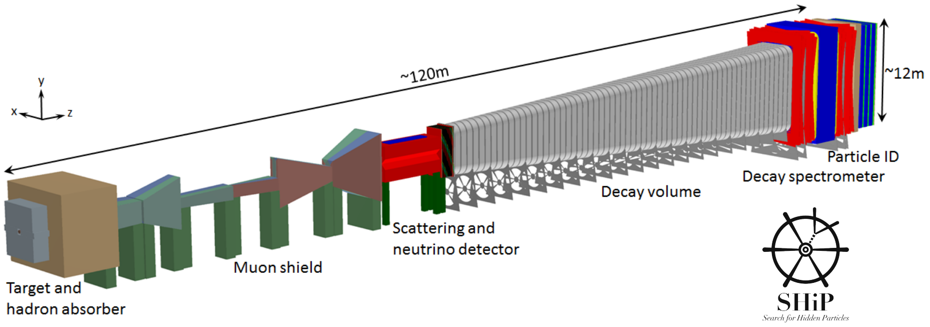


Figure 1. SHiP facility layout [19].

efficiency for charm and beauty mesons and their decay products, while maintaining the lowest possible background rate. The 400 GeV/c proton beam extracted from the SPS will be dumped on a high density W/Mo target with the aim of accumulating 2×10^{20} protons on target during 5 years of operation. The charm production at SHiP will exceed that of any existing or planned facility. The SHiP detector, shown in figure 1, incorporates two complementary apparatuses, the Scattering and Neutrino Detector (SND), and the Decay Spectrometer (DS). The SND will be used to search light dark matter particles, and perform neutrino physics measurements. The DS aims at measuring the visible decays of hidden sector particles by reconstructing their decay vertices in a 50 m long decay volume, making use of a magnetic spectrometer, veto systems and particle identification detectors. Further details of the design of the detector can be found in ref. [19]. Such a setup will allow the SHiP experiment to probe a variety of models that predict light long-lived exotic particles.

Since particles originating in charm and beauty meson decays are produced with a significant transverse momentum with respect to the beam axis,[1] the detector is placed as close as possible to the target. The high flux of muons produced in the target represents a serious background in searches for hidden particles. A critical component of the SHiP experiment is the muon shield [20], which deflects muons produced in the target away from the detector placed downstream of the target. The SHiP detector is designed to reconstruct the exclusive decays of hidden particles and to reduce the background to less than 0.1 events in the full five year period of operation.

3 The SHiP simulation

The simulation of the various physics processes of the response of the SHiP detector are handled by the `FairShip` software suite, which is based on the `FairRoot` software framework [21]. Within `FairShip`, primary collisions of protons are generated with `Pythia8` and the subsequent propagation and interactions of particles are simulated with `GEANT4`. Neutrino interactions are simulated with `GENIE` [22], while heavy flavour production and inelastic muon interactions with `Pythia6` [23] and `GEANT4`. Secondary heavy flavour production in cascade interactions of hadrons originating from the initial proton collision with the SHiP target is also taken into account [24]. The pattern recognition algorithms used to reconstruct tracks from the hits on the strawtubes of the DS are described in [25], and the algorithms for particle identification are presented in [26].

[1] In the SHiP coordinate system the z-axis is along the beam line and the y-axis is pointing upward.

In order to optimise the design of the active muon shield, and develop the reconstruction and selection algorithms of the SHiP experiment, a large simulation campaign was undertaken. Muons produced in the SHiP target were simulated with momentum $p > 10\,\mathrm{GeV}/c$ and a sample corresponding to approximately 10^{11} protons on target was produced. In order to enhance this sample with muons likely to enter the DS, the cross-section of muons produced from decays of ρ^0, ω, η and η' mesons was enhanced by a factor of 100. Similarly, the cross-section for photon conversions into muon pairs was also enhanced by the same factor. The full simulation of this sample, corresponding to a fraction of the 4×10^{13} protons-on-target SPS spill, required months of running on dedicated CPU farms. An order of magnitude increase of this sample could be achieved by exploiting symmetries of the system, such as that in the azimuthal plane of the collision. However, generating even larger samples commensurate to the 2×10^{20} SPS protons on target expected during the lifetime of the experiment is impossible using conventional particle simulation methods. The simulation of the initial proton interaction with the SHiP target, including the subsequent secondary interactions of particles with the target and the hadron absorber, requires significant computing power. Methods such as `SMOTE` [27] and `ADASYN` [28] could be used to synthesise a sample of muons that is larger than the original fully simulated sample. These methods rely on producing muons whose position and momentum vectors take values that lie in between those of existing muons in the fully simulated sample. Generative adversarial networks can offer an alternative way of producing orders of magnitude larger samples with minimal expense to the fidelity of the generated muons.

4 Generative adversarial networks

Neural networks model functions that map an n-dimensional input parameter space into an m-dimensional output, and are widely employed in the particle physics community. A traditional neural network is built up of multiple *layers*: an input layer, one or more intermediate hidden layers, and an output layer. Layers are built from many individual *nodes*, and a pattern of connections joins nodes in adjacent layers. Each node has an associated tunable *bias* term that acts as an activation threshold of the node, and each connection has an associated tunable *weight* representing the strength of the connection. The simplest pattern of connections between layers is one were the nodes in one layer are fully connected with nodes in the adjacent layer. In this configuration the output value of each node is calculated by firstly calculating the sum of the output values of each node from the previous layer, weighted by the strength of each connection. This weighted sum is then shifted by the bias term and passed through an activation function that modulates the output of a node. Depending on the layer that a particular node belongs to, different types of activation functions can be used. For instance, hidden layers often make use of the so called "leaky rectified linear unit" function [29] and in the final layer a sigmoid function could be used to transform the output into a value between 0 and 1. This choice would be appropriate in a binary classification network, whose output is an estimate of whether the input sample originated from one out of two classes of samples.

A neural network must be trained in order for it to successfully approximate a function. The training process involves tuning and updating the weight and bias parameters of the network, with "supervised learning" being the most traditional approach to training. In the first stage of a binary classification problem, labelled data are passed through the network. Output values are then recorded and compared to the true labels through the use of a *loss function*. The loss function provides a quantitative measure of the network's performance on a set of input training samples. A large value of the loss function indicates that the network is unable to distinguish between the two classes of samples. The value of the loss function is used in a process called *back-propagation* to update the weight and bias parameters across the network in an effort to improve the network's performance [30]. Neural networks are trained in steps, where in each step a small *batch* of training data is used, the loss function is then evaluated using this batch of data, and the weight and bias parameters of the network are updated for the next step.

Generative Adversarial Networks (GANs) employ two competing neural networks, one acting as a generator and the other as a discriminator [31]. The generator $\hat{G}$ is trained to map an input vector of random noise z to an output generated vector $G(z;\theta_g)$, where θ_g are parameters of the network and the dimensionality of z is typically larger than that of $G(z;\theta_g)$. The discriminator $\hat{D}$, with trainable parameters θ_d, is trained to map an input vector x to an output prediction $D(x;\theta_d)$, which is a value between 0 and 1. In the study presented in this paper $G(z;\theta_g)$ and x represent the momentum and position vector of the muons. The value of $D(x;\theta_d)$ represents the probability that x originated from the training sample. A value of $D(x;\theta_d)$ closer to 0 indicates that $\hat{D}$ expects the sample to have been generated by $\hat{G}$, whereas if $D(x;\theta_d)$ is closer to 1 $\hat{D}$ is predicting that the sample originated from the training data.

The discriminator and generator networks are trained using an iterative approach. Firstly, the discriminator is trained to distinguish between generated and training samples via a binary crossentropy loss function L_d. This is a common loss function for training classifier networks and is defined as

$$L_d = -[y_{\text{true}} \log(y_{\text{pred}}) + (1 - y_{\text{true}}) \log(1 - y_{\text{pred}})], \tag{4.1}$$

where y_{true} takes the values of 1 or 0 for the training or generated label of the sample respectively, and y_{pred} is the predicted label by the discriminator given by $y_{\text{pred}} = D(x;\theta_d)$. The value of this loss function increases rapidly the further y_{pred} is from the y_{true}. Large values of the loss function bring significant changes in the values of trainable parameters θ_d in the network.

The generator network is then trained in a stacked model which directly connects the output x_{gen} of $\hat{G}$ to the discriminator prediction $D(x_{\text{gen}};\theta_d)$. This is the adversarial component of the GAN, it is only the feedback of $\hat{D}$ that influences the training of $\hat{G}$. The x_{gen} never directly affects the training of $\hat{G}$. In this stacked model all training parameters of the discriminator, θ_d, are fixed to the values obtained from the previous training step of $\hat{D}$. The trainable parameters, θ_g, of the generator are updated based on the loss function, L_g, whose value depends on the output of the discriminator and is defined as

$$L_g = -\log(D(x_{\text{gen}};\theta_d)). \tag{4.2}$$

Low values of $D(x_{\text{gen}};\theta_d)$ indicate that the discriminator is confident that the sample x_{gen} originated from the generator, leading to a large value of L_g. Generated samples that closely resemble training samples will return higher values of $D(x_{\text{gen}};\theta_d)$ and consequently lower values of L_g as the discriminator is successfully tricked into guessing a sample originated from the training sample.

The training of the discriminator and generator is repeated using the samples generated by the previous step. The training of the GAN is completed when generated samples $G(z;\theta_g)$ are indistinguishable from training samples. Different approaches are employed to determine the end of the training. In this paper the metric used to monitor the quality of the training is discussed in section 5.2. The overall accuracy of the generator depends on how well the discriminator is trained to distinguish between generated and training samples.

5 GANs for the SHiP experiment

The GAN is trained on a sample of 3.4×10^8 muons passing through the target and hadron absorber in the full simulation campaign discussed in section 3. As mentioned, this training sample is artificially enriched with muons from rare processes. Therefore, in order to obtain a physical admixture of muons from various sources, batches of muons are extracted from the training sample according to a probability that corrects for this enhancement.

Training is performed on the position $\mathbf{r}$ and momentum $\mathbf{p}$ vectors of these muons at their point of production. Therefore, the GAN generates position and momentum vectors of muons at their production point within the target. Subsequently, they are propagated through the active muon shield and the Decay Spectrometer, relying on `GEANT4` to simulate muon interactions with matter. This procedure allows for a fast production of large muon samples, while maintaining the flexibility to optimise the magnetic shield and downstream detector elements of SHiP, as well as the ability to correct for effects due to the spatial distribution of the proton beam impinging on the target.

Four separate GANs are trained, separated by muon charge and prompt or displaced origin. The x- and y-coordinates of muons originating from prompt decays of mesons such as the ρ^0, ϕ and J/ψ are always the same. This is a consequence of the training sample that relies on `Pythia` with no smearing of the proton-beam distribution. As such, muons from prompt sources are treated separately from muons originating from other sources. Therefore, the GANs trained on prompt muons generate four features $(z, \mathbf{p})$, and the GANs for non-prompt muons generate six $(\mathbf{r}, \mathbf{p})$. In this approach correlations between muons produced in pairs from, for example, vector-meson decays are ignored. Muons are generated individually and any correlation is assumed to be lost via the multiple scattering of the muons through the hadron absorber and muon shield.

5.1 Pre-processing

The distribution of the x- and y- coordinates of muons from non-prompt sources is extremely peaked around the interaction point. Therefore, each value of the x (y) distribution x^i (y^i) is transformed as

$$x^i_{\text{trans}} = \begin{cases} -\sqrt{|x^i - \overline{x}|} & \text{if } x^i < 0, \\ \sqrt{|x^i - \overline{x}|} & \text{if } x^i > 0, \end{cases} \tag{5.1}$$

before training the GANs. This transformation widens the distributions, which proves easier for the GANs to model. The distributions of all the input features are then normalised to values between -1 and 1. This transformation is reversed to obtain physical values of the generated output.

5.2 Figure of merit

An important requirement of the full simulation of the SHiP detector is to accurately model the flux of muons reaching the Decay Spectrometer. This flux crucially depends on the momentum distribution of the muons entering the muon shield of the SHiP experiment. Therefore, muons generated through the GAN approach must closely match the kinematic distributions of the muons produced in the target using the full simulation.

In order to optimise the architecture of the networks and to quantify the quality of the training procedure a figure of merit, FoM, is developed with the following requirements. The FoM must account for how well the GAN is able to model individual features and the correlations between them. Furthermore, it is important that the FoM offers an independent metric of the quality of the training of the GAN since the discriminator and the generator of the GAN improve in tandem during the iterative training procedure. Finally, the calculation of the FoM must be fast so that it does not slow down the training process.

During the training process, small test samples are generated to test the progress of the procedure. As the number of muon features can span a six dimensional space, a small generated sample of muons results in a sparsely populated feature space. Therefore, traditional binned goodness of fit methods, such as χ^2-tests, break down as almost all bins in this space have a low occupancy. Boosted decision trees can overcome this issue [32] and satisfy the aforementioned requirements on the FoM.

A gradient boosted decision tree, BDT, is trained periodically to distinguish between generated and fully simulated muon samples. The BDT uses 100,000 muons generated from the latest GAN configuration and 100,000 randomly selected, fully simulated muons. Half of the muons in each sample are used for training and the other half for testing. The resulting performance of the BDT is quantified through the area under the receiver operating characteristic curve (ROC AUC). A generated sample that is indistinguishable from a fully simulated sample would return an ROC AUC value of 0.5.

5.3 Network optimisation and GAN architecture

All networks are trained with a mini-batch gradient descent approach [33] and at each training step the networks use a sub-sample of training data. The generators and discriminators of the GANs are built using only fully connected layers, resulting in a GAN performance that is independent of the ordering of the muon parameters in the vectors of features that make up the training sample. The number of nodes, the size of the batch sub-sample, the number of layers and the learning rate of the networks are coarsely optimised through a grid search over these parameters. The four GANs are trained until the ROC AUC of the BDT based FoM described in section 5.2 flattens out, and the selected architecture is that which minimises the ROC AUC of the FoM.

As a result of this optimisation procedure, the GANs for both prompt and non-prompt muons follow the architecture shown in figure 2. Leaky rectified linear unit activation functions are used at every hidden layer. The generator and the discriminator have two hidden layers in an inverted pyramidal structure. For the prompt muon GANs, the number of nodes in each hidden layer of $\hat{G}$ are 1536 and 3072 and for $\hat{D}$ are 768 and 1536. For the non-prompt GANs, the number of corresponding nodes are 512 and 1024 for $\hat{G}$ and 768 and 1536 for $\hat{D}$. The input to the generators relies on sampling from a latent space given by a 100 dimensional unit Gaussian distribution. The last layer of $\hat{G}$ has a tanh activation function in accordance to the transformed range of the input features described in section 5.1. The last layer of $\hat{D}$ has a sigmoid activation function providing an output between 0 and 1 that represents $\hat{D}$'s judgement on the origin of a sample. Dropout layers with

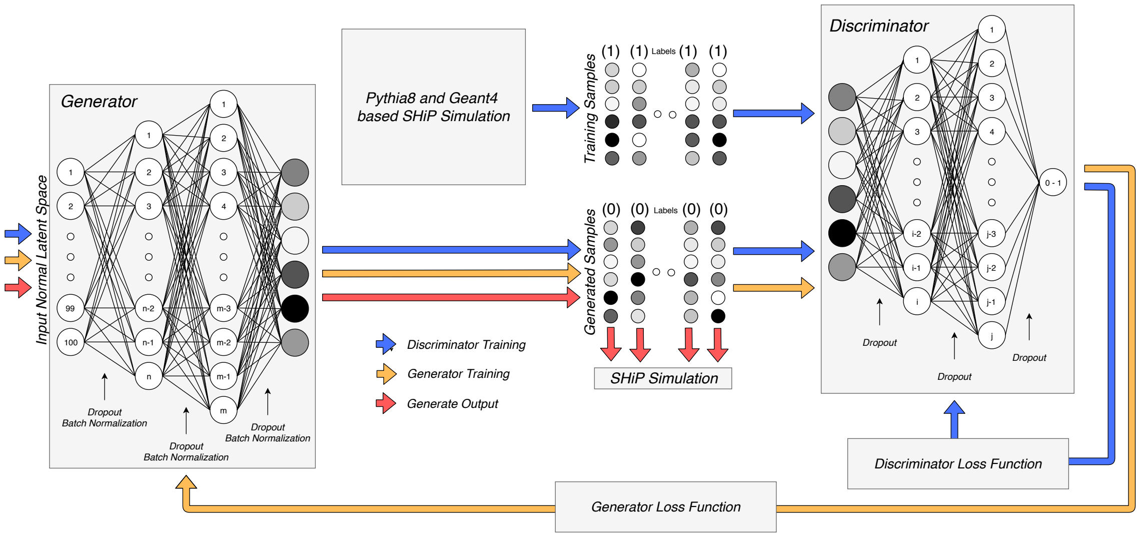


Figure 2. Optimal GAN architecture obtained for the simulation of muon background in SHiP. The number of nodes in each layer for prompt μ and non-prompt μ GANs are given in the text. Arrows indicate the flow of samples and loss information for each stage of training and generation. The features in the generated and training samples can take values between -1 and 1 as denoted by the varying shades of grey.

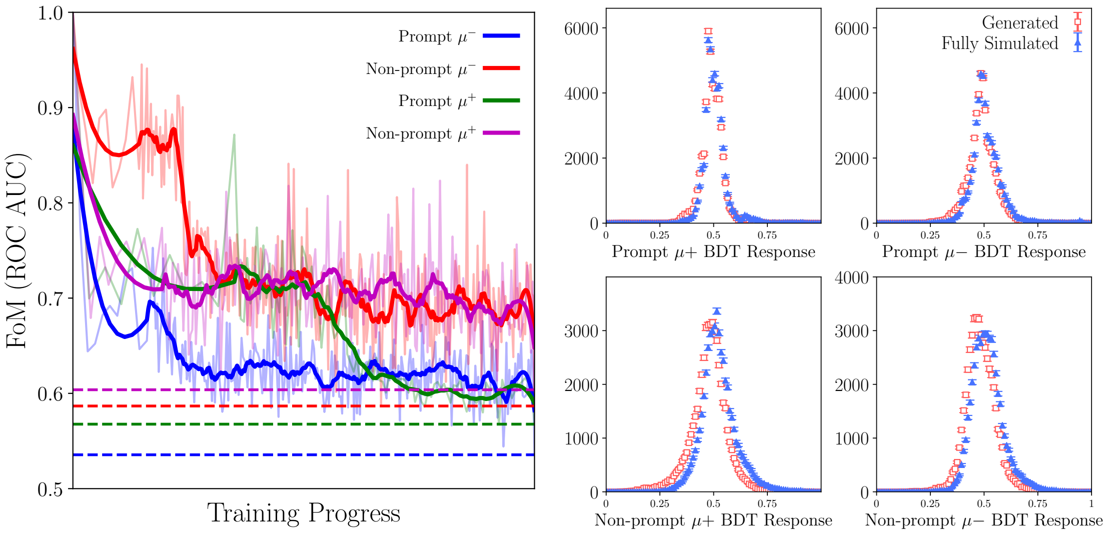
a dropout probability of 0.25 are added between each layer of $\hat{G}$ and $\hat{D}$ to prevent overfitting [34]. Batch normalisation layers are also added between layers of $\hat{G}$ [35].

For this study the `Adam` optimisation algorithm [36] was used in training the networks. Employing the `AMSgrad` algorithm with the `Adam` optimiser increased the stability of our output loss and FoM progress with training [37]. A momentum parameter of `Adam`, β_I, is used with a value of 0.5 to control the progress of the gradient descent during the training of the network.

6 GAN performance

The progress of the FoM throughout the training of each GAN, as well as the BDT distributions of the optimal GAN models are shown in figure 3. The final FoM values for the prompt μ^+ and μ^- GAN models are 0.57 and 0.54 respectively. Whereas, the non-prompt μ^+ and μ^- GAN models return FoM values of 0.60 and 0.59 respectively. A more sophisticated optimisation procedure of the network architecture, such as that suggested in [38], could result in an even better GAN performance.

In order to further visualise the level of agreement between the generated and fully simulated samples, a physical sample of GAN based muons is produced by combining the output from each of the four generators according to the expected production fractions of prompt and non-prompt muons in the simulation. Figure 4 compares the one- and two-dimensional distributions of each unique pair of features between fully simulated and generated muons. The GANs can overall reproduce the correct correlations between features, although the tails of the (x, y, z) position distributions are underestimated.



(a)

(b)

Figure 3. (a) Progress of the FoM ROC AUC value throughout the training of all 4 GANs, raw and smoothed data is displayed. Dashed lines indicate the FoM AUC ROC values of models chosen to generate muons in this paper. Although models were trained past this point this was the lowest FoM AUC ROC value obtained, (b) Distributions of the figure of merit BDT response for both fully simulated and GAN-based muon samples for prompt and non-prompt μ^- and μ^+.

Modelling of the momentum (p) and transverse momentum (p_T) plane accurately is crucial in order to obtain the correct flux of muons reaching the SHiP Decay Spectrometer. Figure 5 compares the (p, p_T) plane between the fully simulated and generated samples. The GANs can largely reproduce the correlations between these features, however they particularly underestimate the number of muons with $p_\mathrm{T} > 3\,\mathrm{GeV}/c$. The effect of this mismodelling on the rate and kinematics of muons reaching the Decay Spectrometer is discussed in section 7. To correct the momentum distribution of the generated muons, the three-dimensional (p_x, p_y, p_z) distributions of the fully simulated and generated muons are each fit using a three-dimensional Kernel Density Estimator, for example see ref. [39]. For each generated muon, an individual correction weight is derived by taking the ratio between fully simulated over the fully generated KDEs at the corresponding (p_x, p_y, p_z) muon coordinate.

This generative approach accurately models the production of muons in the SHiP target, as long as the kinematic distributions of the muons lie within the phase space covered by the fully simulated training sample. Therefore, samples of muons produced through the GAN are designed to compliment, rather than replace, existing fully simulated samples. By generating vast samples of muons through this generative approach, a better understanding of the performance of the muon shield and the detector response for muons that lie within the kinematic region of the fully simulated sample can be obtained.

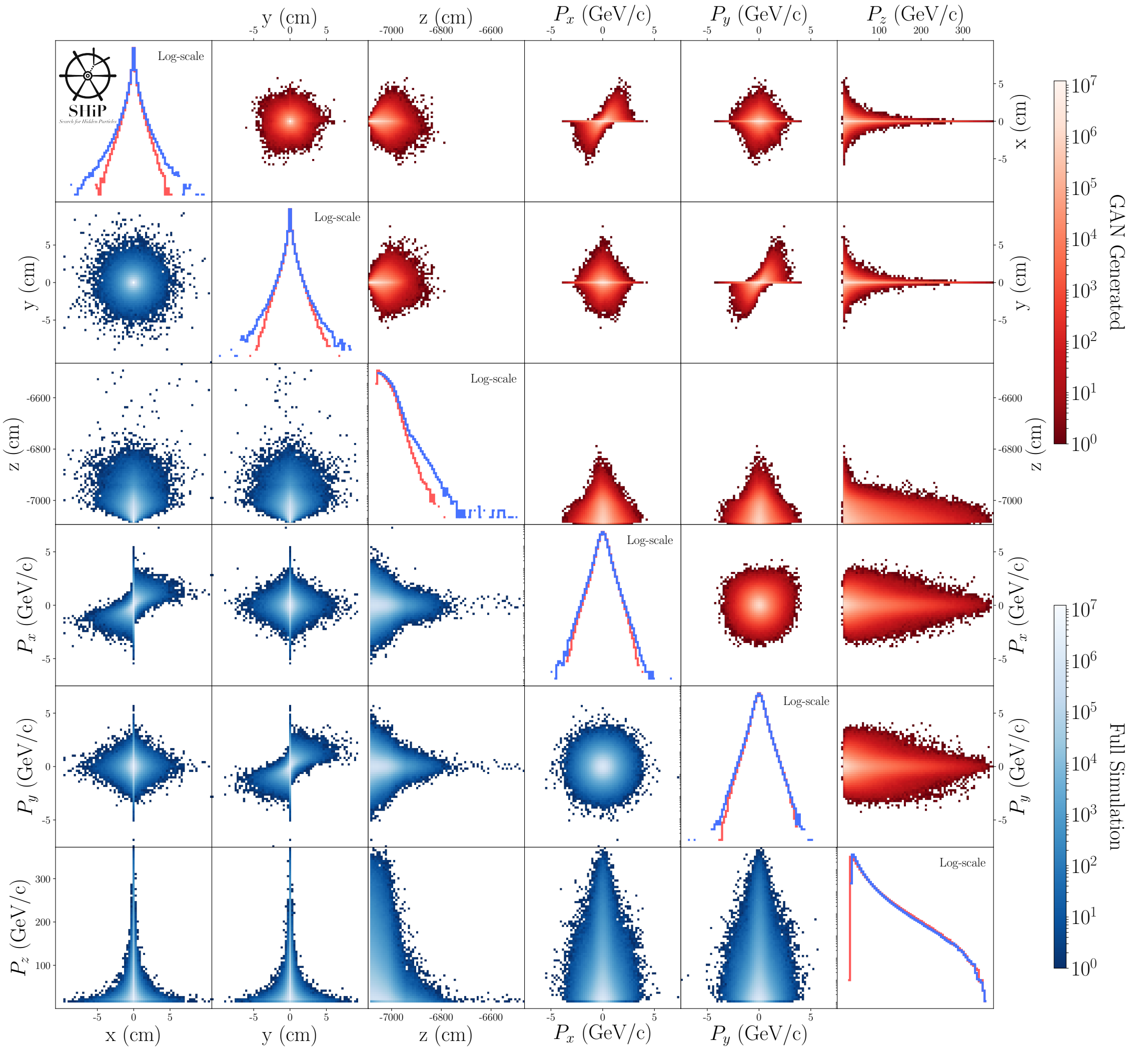


Figure 4. Two-dimensional distributions of all unique combinations of muon features for GAN based (upper-half) and fully simulated (lower-half) muons produced in the SHiP target. One-dimensional log scale comparisons of each feature are presented along the diagonal.

7 Reconstructing GAN generated muons

The generated muons are processed using `FairShip` to simulate their passage through the magnetic shield and the response of the downstream SHiP detector. Figure 6 shows the p vs p_{T} distribution of reconstructed muon tracks in the Decay Spectrometer of SHiP resulting from the GAN based muon sample. A comparison to the reconstructed muon tracks originating from the full simulation sample is also shown. The effect of the residual correction to the kinematics of the GAN based muon sample discussed in section 6 is found to have a small effect.

Figure 7 shows the momentum distributions at the production point of the muons, for muons that are reconstructed in the DS. The GAN based and fully simulated muons display similar features in the p vs p_{T} plane. The fully simulated sample exhibits localised hot-spots. These are due to the use of event weights that account for enhancement factors of particular processes that give rise to muons likely to enter the Decay Spectrometer as discussed in section 3.

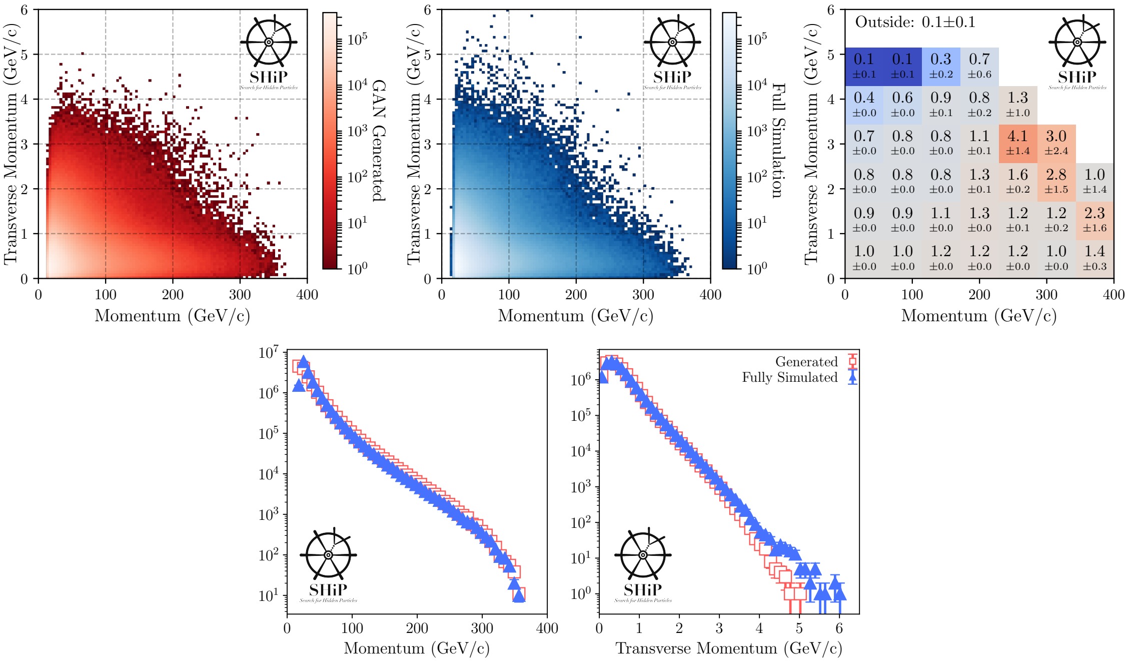


Figure 5. Two-dimensional p vs p_T distributions for GAN based (top-left), fully simulated (top-middle) and the ratio (top-right) of muons produced in the SHiP target. The comparisons of the one dimensional projections for p (bottom-left) and p_T (bottom-right) are also shown.

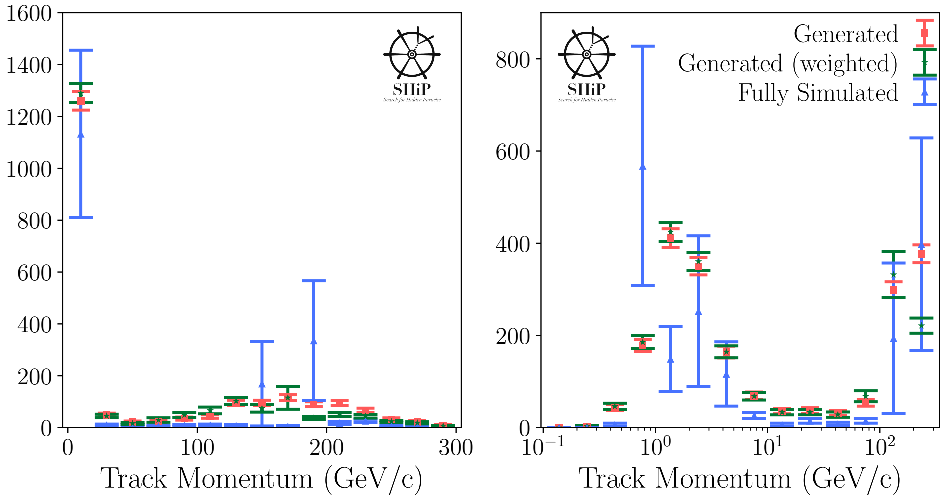


Figure 6. Distribution in linear (left) and log-scale (right) of the reconstructed track momentum of muons in the Decay Spectrometer. The distributions of both GAN based and fully simulated muons are also shown together with the effect of the correction to the residual mismodelling of the muon kinematics from the GAN based sample. The distributions are normalised such that they correspond to the same number of protons on target.

The rate of muons that survive the magnetic shield and are reconstructed in the Decay Spectrometer is given in table 1. Both the full rate, and the rate of muons with an initial (p, p_T) distribution corresponding to the upper region of figure 7 agree when comparing the GAN based and fully simulated muon samples. The correction to the kinematic distributions of the GAN based muons discussed in section 6, changes the rate of generated muons entering the Decay Spectrometer by ~ 4%.

Table 1. Rates of reconstructed muons in the Decay Spectrometer. The uncertainty on the GAN based muons reflects the statistical uncertainty of the generated muon sample, given the model described in section 6.

Approach	Full Rate (kHz)	Upper Region Rate (kHz)
Full simulation	13.9 ± 3.4	4.7 ± 2.2
GAN	15.8 ± 0.3	5.5 ± 0.2
GAN (weighted)	15.2 ± 0.5	4.7 ± 0.4

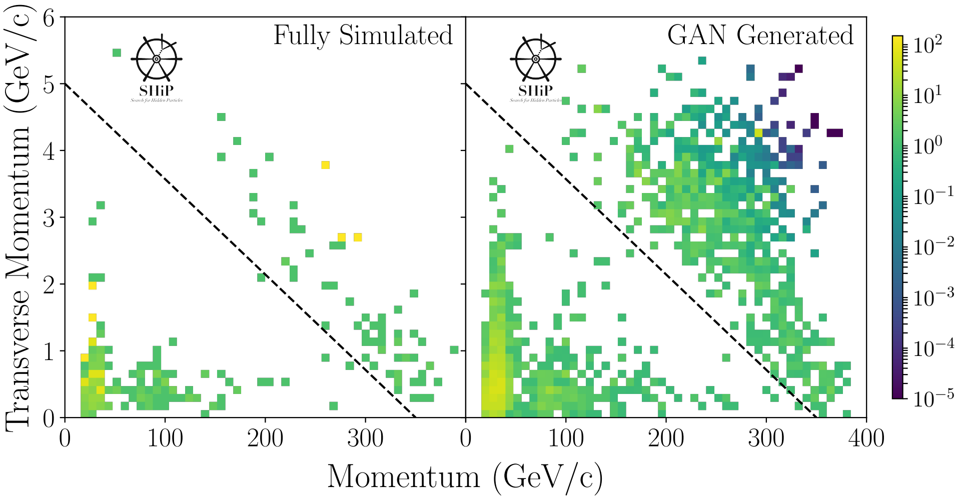


Figure 7. Initial momentum of muons passing through the SHiP active muon shield with well reconstructed tracks in the Decay Spectrometer. Full simulation data is presented on the left and generated data on the right. The dashed line indicates the upper region analysed in table 1.

8 Benchmarking

With a small expense in the fidelity between the generated and fully simulated sample, the generative approach can produce samples of muons at greater speed. Generating samples of muons from GANs on a GPU provides a speed-up of $O(10^6)$ relative to the full `Pythia8` and `GEANT4` proton-on-target simulation. This test was performed using `Keras`(v2.1.5) on a `TensorFlow` backend (v1.8.0) on a single `Nvidia Pascal P100` GPU card. This speed-up factor includes all the computations required to transform the output features of the generator into physical values. Generating muons using the GAN approach on a CPU is an order of magnitude slower than on a GPU.

Table 2 summarises the results of this performance test. The gain in speed using the generative approach is partly due to the small production cross-section of muons with $p > 10\,\mathrm{GeV}/c$, requiring $O(10^3)$ proton-on-target interactions to be simulated through `Pythia8` in order to generate a single muon.

Table 2. Summary of benchmarking results.

Target simulation method	Muons produced in 5 minutes	Time to simulate single muon (s)
Pythia8 and GEANT4	~ 1	1.1×10^{-1}
GAN (CPU)	7.5×10^{5}	4.0×10^{-4}
GAN (GPU)	3.5×10^{6}	8.6×10^{-5}

9 Conclusion

This paper demonstrates the success of using a modern machine learning method to approximate the output of a complex and computationally intensive simulation of muons originating from SPS protons impinging on the target of the SHiP experiment. The GAN models presented in this paper produce samples that emulate the characteristics of the fully simulated sample and can approximate the kinematic correlations of the muons produced in the SHiP target. Furthermore, muons generated by these GANs correctly describe the expected flux and kinematic distributions of muons that survive the magnetic shield and are reconstructed in the Decay Spectrometer of the SHiP detector.

The generative models developed in this paper can produce muons $O(10^6)$ times faster than the current Pythia8 and GEANT4 simulation of the SHiP target. However, the muons produced by the generative model are only representative of regions of phase space populated by the full simulation of the target. These generative models are not capable of accurately extending the tails of their training distributions, and are not intended to replace the fully simulated background sample. Generated muons can be used in parallel to complement ongoing background and detector optimisation studies, where this approach can offer a vast increase in the sample size of statistically limited muon background studies at SHiP. Finally, the generative approach presented in this paper can be used to produce muons according to a model trained directly on real data, such as that from the recent muon-flux beam-test campaign of the SHiP collaboration [40]. Such an approach circumvents the challenge of tuning the multitude of parameters that control the simulation in order to match the data.

Acknowledgments

This work was carried out using the computational facilities of the Advanced Computing Research Centre, University of Bristol — http://www.bristol.ac.uk/acrc/. We would like to thank the NVIDIA corporation for the donation of a Titan Xp which was used for this research.

The SHiP Collaboration wishes to thank the Castaldo company (Naples, Italy) for their contribution to the development studies of the decay vessel. The support from the National Research Foundation of Korea with grant numbers of 2018R1A2B2007757, 2018R1D1A3B07050649, 2018R1D1A1B07050701, 2017R1D1A1B03036042, 2017R1A6A3A01075752, 2016R1A2B4012302, and 2016R1A6A3A11930680 is acknowledged.

The support from the FCT — Fundação para a Ciência e a Tecnologia of Portugal with grant number CERN/FIS-PAR/0030/2017 is acknowledged. The support from the Russian Foundation for Basic Research (RFBR), grant 17-02-00607, the support from the TAEK of Turkey, and the support from the U.K. Science and Technology Facilities Council (STFC), grant ST/P006779/1 are acknowledged.

We thank M. Al-Turany, F. Uhlig. S. Neubert and A. Gheata their assistance with FairRoot. We acknowledge G. Eulisse and P.A. Munkes for help with Alibuild.

We thank M. Daniels for his contributions to the construction of the liquid-scintillator testbeam detectors.

The muon flux and charm cross section measurements this summer would not have been possible without a significant financial contribution from CERN. In addition, several member institutes made large financial and in-kind contributions to the construction of the target and the spectrometer sub detectors, as well as providing expert manpower for commissioning, data taking and analysis. This help is gratefully acknowledged.

References

[1] H. Zhang et al., *Stackgan: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks*, arXiv:1612.03242.

[2] Y. Pu et al., *Variational Autoencoder for Deep Learning of Images, Labels and Captions*, in *Advances in neural information processing systems. Proceedings of Neural Information Processing Systems 2016*, Barcellona Spagna (2016) [arXiv:1609.08976].

[3] C. Ledig et al., *Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network*, *IEEE Conf. Comput. Vis. Pattern Recognit.* **2017** (2017) 1 [arXiv:1609.04802].

[4] T. Wang et al., *High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs*, *IEEE Conf. Comput. Vis. Pattern Recognit.* **2018** (2018) 1 [arXiv:1711.11585].

[5] P. Isola, J. Zhu, T. Zhou and A.A. Efros, *Image-to-Image Translation with Conditional Adversarial Networks*, arXiv:1611.07004.

[6] GEANT4 collaboration, *GEANT4: A Simulation toolkit*, *Nucl. Instrum. Meth.* **A 506** (2003) 250.

[7] HEP Software Foundation collaboration, *A Roadmap for HEP Software and Computing R&amp;D for the 2020s*, *Comput. Softw. Big Sci.* **3** (2019) 7 [arXiv:1712.06982].

[8] P. Canal et al., *GeantV: from CPU to accelerators*, PoS(ICHEP2016)177.

[9] M. Mirza and S. Osindero, *Conditional Generative Adversarial Nets*, arXiv:1411.1784.

[10] L. de Oliveira, M. Paganini and B. Nachman, *Learning Particle Physics by Example: Location-Aware Generative Adversarial Networks for Physics Synthesis*, *Comput. Softw. Big Sci.* **1** (2017) 4 [arXiv:1701.05927].

[11] M. Erdmann, J. Glombitza and D. Walz, *A Deep Learning-based Reconstruction of Cosmic Ray-induced Air Showers*, *Astropart. Phys.* **97** (2018) 46 [arXiv:1708.00647].

[12] M. Paganini, L. de Oliveira and B. Nachman, *CaloGAN: Simulating* 3*D high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks*, *Phys. Rev.* **D 97** (2018) 014021 [arXiv:1712.10321].

[13] D. Derkach, N. Kazeev, F. Ratnikov, A. Ustyuzhanin and A. Volokhova, *Cherenkov Detectors Fast Simulation Using Neural Networks*, in *10th International Workshop on Ring Imaging Cherenkov Detectors (RICH 2018)*, Moscow Russia (2018) [arXiv:1903.11788].

[14] A. Butter, T. Plehn and R. Winterhalder, *How to GAN LHC Events*, arXiv:1907.03764.

[15] S. Otten et al., *Event Generation and Statistical Sampling for Physics with Deep Generative Models and a Density Information Buffer*, arXiv:1901.00875.

[16] B. Hashemi, N. Amin, K. Datta, D. Olivito and M. Pierini, *LHC analysis-specific datasets with Generative Adversarial Networks*, arXiv:1901.05282.

[17] SHiP collaboration, *A facility to Search for Hidden Particles (SHiP) at the CERN SPS*, arXiv:1504.04956.

[18] T. Sjöstrand, S. Mrenna and P.Z. Skands, *A Brief Introduction to PYTHIA 8.1*, *Comput. Phys. Commun.* **178** (2008) 852 [arXiv:0710.3820].

[19] SHiP collaboration, *SHiP Experiment — Progress Report*, CERN-SPSC-2019-010 (2019).

[20] SHiP collaboration, *The active muon shield in the SHiP experiment*, 2017 *JINST* **12** P05011 [arXiv:1703.03612].

[21] M. Al-Turany et al., *The FairRoot framework*, *J. Phys. Conf. Ser.* **396** (2012) 022001.

[22] C. Andreopoulos et al., *The GENIE Neutrino Monte Carlo Generator*, *Nucl. Instrum. Meth.* **A 614** (2010) 87 [arXiv:0905.2517].

[23] T. Sjöstrand, S. Mrenna and P.Z. Skands, *PYTHIA 6.4 Physics and Manual*, *JHEP* **05** (2006) 026 [hep-ph/0603175].

[24] H. Dijkstra and T. Ruf, *Heavy Flavour Cascade Production in a Beam Dump*, CERN-SHiP-NOTE-2015-009 (2015).

[25] E. Van Herwijnen, T. Ruf, M. Ferro-Luzzi and H. Dijkstra, *Simulation and pattern recognition for the SHiP Spectrometer Tracker*, CERN-SHiP-NOTE-2015-002 (2015).

[26] B. Hosseini and W.M. Bonivento, *Particle Identification tools and performance in the SHiP Experiment*, CERN-SHiP-NOTE-2017-002 (2017).

[27] N.V. Chawla, K.W. Bowyer, L.O. Hall and W.P. Kegelmeyer, *SMOTE: synthetic minority over-sampling technique*, *J. Artif. Intell. Res.* **16** (2002) 321 [arXiv:1106.1813].

[28] H. He, Y. Bai, E.A. Garcia and S. Li, *ADASYN: Adaptive synthetic sampling approach for imbalanced learning*, *IEEE Int. Joint Conf. Neural Netw.* **2008** (2008) 1

[29] A.L. Maas, A.Y. Hannun and A.Y. Ng, *Rectifier Nonlinearities Improve Neural Network Acoustic Models*, *Proc. ICML* **30** (2013) 3.

[30] D.E. Rumelhart, G.E. Hinton and R.J. Williams, *Learning representations by back-propagating errors*, *Cognitive Modeling* **5** (1988) 1 [*Nature* **323** (1988) 533].

[31] I.J. Goodfellow et al., *Generative Adversarial Networks*, arXiv:1406.2661.

[32] C. Weisser and M. Williams, *Machine learning and multivariate goodness of fit*, arXiv:1612.07186.

[33] S. Ruder, *An overview of gradient descent optimization algorithms*, arXiv:1609.04747.

[34] N. Srivastava et al., *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*, *J. Mach. Learn. Res.* **15** (2014) 1929.

[35] S. Ioffe and C. Szegedy, *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*, arXiv:1502.03167.

[36] D.P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, arXiv:1412.6980.

[37] S.J. Reddi, S. Kale and S. Kumar, *On the Convergence of Adam and Beyond*, arXiv:1904.09237.

[38] M. Jaderberg et al., *Population Based Training of Neural Networks*, arXiv:1711.09846.

[39] D.W. Scott, *Multivariate density estimation: theory, practice, and visualization*, John Wiley & Sons, New York U.S.A. (2015).

[40] SHiP collaboration, *Alignment of the muon-flux spectrometer in FairShip using the survey measurements*, CERN-SHiP-NOTE-2019-001 (2019).