Received July 12, 2021, accepted July 25, 2021, date of publication August 2, 2021, date of current version August 10, 2021.

*Digital Object Identifier 10.1109/ACCESS.2021.3101946*

# High Energy Physics Calorimeter Detector Simulation Using Generative Adversarial Networks With Domain Related Constraints

**GUL RUKH KHATTAK[1,2], SOFIA VALLECORSA[1], FEDERICO CARMINATI[1], AND GUL MUHAMMAD KHAN[2]**

[1]CERN European Organization for Nuclear Research, 1211 Geneva, Switzerland
[2]Department of Electrical, University of Engineering and Technology Peshawar, Peshawar 25000, Pakistan

Corresponding author : Gul Rukh Khattak (gul.rukh.khattak@cern.ch)

**ABSTRACT** Generative Adversarial Networks (GANs) have gained notoriety by generating highly realistic images. The present work explores GAN for simulating High Energy Physics detectors, interpreting detector output as three-dimensional images. The demands and requirements of a scientific simulation are quite stringent, as compared to the domain of visual images. Image characteristics such as pixel intensity and sparsity, for example, have very different distributions. Moreover, detector simulation requires conditioning on physics inputs, and domain knowledge becomes essential. We, therefore, adjust the pre-processing and incorporate physics-based constraints in the loss function. We also introduce a multi-step training process based on transfer learning by breaking up the task complexity. Validation of the results primarily consists of a detailed comparison to full Monte Carlo in terms of several physics quantities where a high level of agreement is found (ranging from a few percent up to 10% across a large particle energy range). In addition, we assess the performance by physics unrelated metrics, thereby proving further the variability and pertinence through diverse standpoints. We have demonstrated that an image generation technique from vision can successfully simulate highly complex physics processes while achieving a speedup of more than three orders of magnitude in comparison to the standard Monte Carlo.

**INDEX TERMS** 3D vision, fast simulation, generative adversarial networks, high energy physics, image processing and generation, transfer learning.

## I. INTRODUCTION

Simulation of particle transport through matter is fundamental for interpreting the results of High Energy Physics (HEP) experiments. The particles undergo complex interactions while traversing the detector material with stochastic outcomes. The modeling of these processes is carried out with the help of Monte Carlo (MC) techniques that rely on repeated random sampling. The MC simulation meets the theoretical predictions with a high degree of precision but is both time and resource intensive. The Worldwide LHC Grid [1] has currently more than 50% of its resources devoted only to simulation [2]. The future High Luminosity LHC [3] will require 100 times more simulated data, thus surpassing the expected resource availability. The HEP community is therefore highly motivated to explore fast alternatives, often trading some accuracy for speed, if only partially, for certain applications. Fast simulation is a set of established techniques that replaces parts of the detailed MC simulation with alternative approaches. Currently parametrized approaches [4]–[6] or lookup tables [7] can provide between 10 and 100 times speedup, achieving different levels of accuracy. Here we investigate an alternative approach based on Deep Neural Networks.

HEP detectors can be described as 3D cameras, recording pictures of particle collisions. Calorimeters, in particular, detect particles by measuring the energy deposited in interactions with matter. Segmented calorimeters consist of alternate

The associate editor coordinating the review of this manuscript and approving it for publication was Shadi Alawneh.

arrays of active sensor material and passive dense layers, to ensure that the incoming (primary) particle will deposit most of its energy inside their volume. The energy depositions in calorimeter cells can be compared to the monochromatic pixel intensities of a 3D image.

The 3DGAN is a convolutional GAN architecture aimed at stimulating the calorimeter's energy response. The calorimeter is a bottleneck in most HEP simulation pipelines due to a large number of complex interactions, taking about 90% of simulation time for some experiments [8]. The model was developed in several steps. The initial proof of concept was a simplified prototype demonstrating successful detector simulation, conditioned by the energy of the primary particle, entering the detector perpendicularly to its surface [9]. This represented a simplification, given that in real-life conditions, particles usually hit the detectors from different directions. The model was revised to learn a joint probability distribution of both the incident particle energy and its direction (quantified by the incident angle). At this stage, we also increased the image size and added new domain-related features (details in Section IV) in order to improve the accuracy of the results. Some preliminary results for this configuration were presented in [10], where the training was run for a small number of epochs and the result validation was limited. The current work refines the results and describes the architecture, training process, and pre-processing in more detail. The physics-based analysis of the results is presented in greater detail along with additional performance studies from other viewpoints. These include classification and regression results by a third-party network pre-trained on the same data set, as well as a more detailed investigation using image quality metrics as a function of input conditions.

Our work on 3DGAN is relevant both to the field of computer vision and image processing as well as High Energy Physics. It represents an example of a successful application of algorithms designed for image processing, to scientific simulation, exhibiting very different characteristics and requiring a higher level of accuracy. It is a fast alternative to detailed Monte Carlo simulation that results in a speedup greater than three orders of magnitude, which would be difficult to achieve using other approaches. The work also demonstrates the possibility of adding domain related constraints to the process of image generation and represents a successful example of transfer learning. The simulation of detector output poses an interesting problem since the stochastic nature of particle interactions results in unique images, yet unlike most vision applications these images exhibit well-defined distributions for observable physics features. Thus both the realism and diversity of the simulated data can be assessed on the basis of these features.

This paper is organized as follows. Section II will briefly review related work focusing on HEP applications. It will be followed by a description of the training data set and the features used for the evaluation of the results in Section III. The current approach will then be presented in Section IV, together with details on the loss function and the network architecture. The main design choices made during the development process will also be discussed. Validation of the results from a physics perspective is accomplished by a detailed comparison of GAN and Monte Carlo as presented in Section V. This section also includes an investigation of additional validation metrics (e.g. structural similarity index) inspired from the image processing domain. We will conclude by summarizing our main contributions and suggestions for future work in Section VI.

## II. PREVIOUS WORK

Generative models represent a fundamental part of deep learning. Over the years this field has seen developments in the Generative Stochastic Networks [11], to the Variational AutoEncoders [12], and Generative Adversarial Networks (GAN) [13]. In particular, Generative Adversarial Networks can successfully generate sharp and realistic images with high resolution [13]. Inspired by the Game Theory [14], adversarial training is defined as a competition between two players: a generator and a discriminator. The discriminator distinguishes real from fake images while the generator tries to fool the discriminator by producing an output as realistic as possible. The process eventually results in the generator learning the distribution of the real data if given enough representation capacity and time.

There are many variants of the GAN methodology, such as WGAN [15], StackGAN [16], ProgressiveGAN [17] etc., demonstrating the generation of high quality and high resolution images. The GAN does not rely on the explicit computation of probability densities and is thus suitable for a wider range of applications. Presently this approach has been applied to problems from many domains: ranging from generating musical notes [18], to natural language [19], to medical data [20], [21], to natural scenes [13] and image denoising [22]. The adversarial approach has also been successfully applied to anomaly detection tasks for industrial [23] and medical [24] applications. Similarly, scientific simulation is another domain where GAN has shown immense success. Astro-physics [25], [26], micorbiology [27], and material composition [28], etc., are some of the avenues explored.

The LAGAN [29] and CALOGAN [30] models introduce the idea of using GAN for High Energy Physics shower simulation as two-dimensional images for a simplified calorimeter use case. Since then, there have been other demonstrations employing deep learning for HEP calorimeter simulation [31]–[33]. The 3DGAN takes inspiration from the ACGAN [34] approach but conditions the input on a set of continuous variables, combined with additional constraints to improve physics accuracy [9], [10]. A recent work [35] takes a different approach and introduces the model-assisted GAN, where the simulation is conditioned by a set of model parameters. The GAN learns the true distribution of model parameters from experimental data and applies this information to the external simulator to reproduce the images more accurately, as well as train a neural network (the emulator) to mimic the simulator response. The approach is verified for two simple test cases involving 2D images. It is a preliminary study and has yet to prove if the technique can withstand the complexity of a realistic HEP simulation. The approach is not yet applicable to the high-granularity future calorimeter (simulated by the 3DGAN) where the experimental data is not available. The 3DGAN input parameters include only the simulation conditions sampled from a uniform distribution, whereas the physics features are learned implicitly that we believe to be a more powerful approach and requires a single network (the generator) for inference, instead of utilizing a generator to generate the distribution of the model features and an emulator to generate the simulation output. Apart from these differences, the time-consuming Monte Carlo simulation, the large range of the input variables, the large size of the 3D images, and the training data are not suitable for the lengthy training setup of the approach. Unlike the model-assisted GAN, the 3DGAN predicts the input conditions from the generated images as feedback and also imposes hard domain-related constraints as deterministic functions of the images that are crucial to achieving the level of accuracy required. The 3DGAN [9], [10] is the first application for 3D convolutions to simulate a high granularity calorimeter. There is a similar later effort [36] employing 3D convolutions, although for only orthogonally incident particles coming with energy from a limited range and with smaller image dimensions.

The novelty of our work resides in the high granularity (high spatial resolution) of the detector we simulate and in the use of three-dimensional convolutions that are essential to preserving all spatial correlations between pixels. Our particular pre-processing, loss function, and two-step training results in high accuracy for a more complex scenario in comparison to other GAN applications for HEP calorimeter simulation, involving a wider range of input variables used for conditioning.

## III. DATA SET

The current work is based on the detailed Monte Carlo simulation data for the proposed high granularity Linear Collider Detector (LCD) [38] geometry, designed in the context of the Compact Linear Collider (CLIC) accelerator [39]. The data are generated using the GEANT4 toolkit [40] that is the state of the art platform for the simulation of particle transport through matter (based on the Monte Carlo methods) and the most widely used package in the particle physics community. Figure 1 presents the proposed detector design with a highly segmented calorimeter. We limit our study to the electromagnetic calorimeter (ECAL) located in the cylindrical region of the LCD calorimeter: a grid of 5.1 $\text{mm}^3$ cells, with an inner radius of 1.5 mm and 25 concentric layers. The data include several particle types (i.e. electrons, photons, neutral, and charged pions) and is publicly available on Zenodo at https://zenodo.org/communities/mpp-hep.

The incoming particle creates an avalanche of secondary particles as it traverses the detector thus generating a characteristic energy deposit pattern (called a ''shower''). Each entry or event in the data set corresponds to an array of cells centered around the barycenter of the particle shower as a three dimensional $51 \times 51 \times 25$ pixelized image. The pixel intensities of this image are the energy depositions for the calorimeter cells. The energy ($E_P$) and the incident angle ($\theta$) of the original particle (''primary particle'') are also stored with each event. The data consist of particles with the $E_P$ range of 2 GeV to 500 GeV and $\theta$ range of 60° to 120°. The training uses 137, 342 electron events with $E_P$ restricted to the range of 100 GeV to 200 GeV for the first step and 400, 000 events with $E_P$ from 2 GeV to 500 GeV for the second training step. The data is divided into a ratio of nine to one for train and test sets. While additional data is used for the validation and detailed analysis presented in Section V (filtered for specific energies and angles from sets of 50, 000 and 100, 000 events respectively).

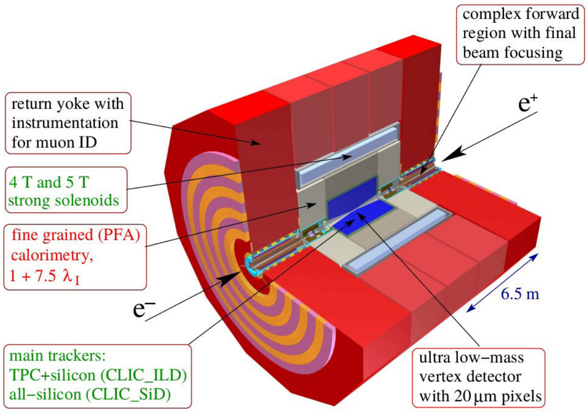

complex forward region with final beam focusing
return yoke with instrumentation for muon ID
e+
4 T and 5 T strong solenoids
fine grained (PFA) calorimetry, 1 + 7.5 λ_I
e−
6.5 m
main trackers: TPC+silicon (CLIC_ILD) all−silicon (CLIC_SiD)
ultra low−mass vertex detector with 20 µm pixels

**FIGURE 1.** **Schematic diagram for the CLIC calorimeter [37].**

The physical characteristics of a particle shower are defined by the underlying physics processes, the particle type, its energy, and incident angle. The 2D projections on different planes can be used to convey visual information.[1] The geometry of a shower is essential information for particle type and energy identification. Frequently used geometrical features consist of shower shapes, moments, and ratio of energy deposited in different parts of a shower. The shower shapes are represented by the energies deposited along different axes. Shower moments are another way of defining the shower geometry. The first moment ($M1$) corresponds to shower center, the second moment ($M2$) to shower width, and similarly higher moments to higher-order features. The fraction of energy deposited in different parts of the shower is studied by dividing the shower into three parts and comparing the fraction of total energy deposited in the first eight cells ($R_1$), the middle nine cells ($R_2$), and the last eight cells ($R_3$).

Each event consists of a shower surrounded by a largely sparse region (typically less than 20% cells in a shower receive some energy deposits). We measure the sparsity as the fraction of cells with deposited energy above a certain

[1] The $Z$ axis lies along the detector depth and $X$, $Y$ are the transverse axes.

threshold $(1 - S)$. The number of cells above a threshold is also denoted by hits $(H)$. The ratio of the total recorded energy $(E_{sum})$ in the active layers of the calorimeter and the corresponding $E_P$ represents the "sampling fraction" $(SF)$. It defines the calorimeter response and is a characteristic of both the detector itself and of the incoming primary particle.

## IV. THE 3D CONVOLUTIONAL GAN

The 3DGAN represents the first proof of concept for the possibility of using 3D convolutional GANs to simulate high granularity calorimeters. Our network is inspired from the Auxiliary Classifier GANs (ACGAN) [34] and InfoGAN [41] concepts. These architectures are a natural extension of the GAN approach and feature a faster convergence and more stable performance by introducing auxiliary tasks for the discriminator. The GAN application to simulation requires the generation of images conditioned on a set of continuous inputs, thus introducing auxiliary tasks such as regression on the conditioning variables, not only stabilize the training but also provide feedback on the conditioning. With the advent of representation learning through deeper models, domain knowledge is often not required [42]. We present an example of combining deep learning with domain related constraints since scientific simulations must conform to scientific laws.

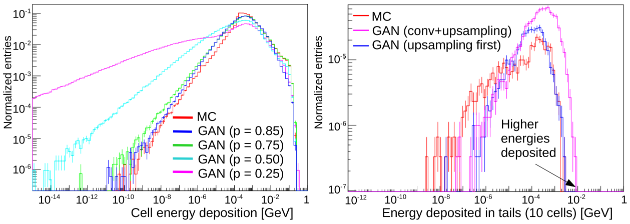


**FIGURE 2. The cell energy distribution for Monte Carlo events vs. GAN generated events. (Left) pre-processing by taking the power of pixel intensities for: p = 0.85 (blue); p = 0.75 (green); p = 0.5 (cyan); p = 0.25 (magenta). (Right) the distribution of the highly sparse region near the periphery along the transverse dimensions (*X* and *Y* axes) for generator with: alternative upsampling and convolutional layers (magenta); upsampling before the convolutions (blue).**

### A. PRE-PROCESSING

One of the challenges in applying image generation techniques to the simulation of detector output lies in the large dynamic range of the deposited energies in detector cells as compared to pixel intensities in a typical RGB image. The cell energy deposits can vary over a large range spanning more than ten orders of magnitude as shown in Figure 2 (right). We explored different pre-processing procedures aimed at reducing this dynamic range. The initial tests using the log of the pixel intensity yielded unsatisfactory results (highly distorted images). We also experimented with the power function of pixel intensities using an exponent $(p)$ smaller than one. We observed that a smaller exponent means faster convergence but greater distortion. Figure 2 (right panel) compares the pixel intensity distribution for different values of the exponent. An optimum value of 0.85 improve the convergence while retaining accuracy at both ends of the spectrum. The generated images can be transformed back to the original range by simply taking the inverse of the power function.

### B. THE ARCHITECTURE

The design of the 3DGAN architecture required a long and tedious process of trial and error, as well as conventions and suggestions from past efforts. The long training times and practically unlimited choice of architectural hyper-parameters do not allow an exhaustive search of the entire design space, although the model development and optimization involved extensive investigation of different model parameters. The 3DGAN initial prototype [9] architecture was inspired by the DCGAN [43] architecture: employing four convolutional layers in both the generator and the discriminator networks. The upsampling layers followed the first two convolutional layers in the generator.

Figure 3 shows the final optimized architecture for the discriminator $(D)$ and the generator $(G)$. The latent space is a vector of 254 random numbers drawn from a Gaussian distribution. The input conditions $E_P$ and $\theta$ are concatenated to this latent vector to create the generator input. The initial prototype [9] used a Hadamard product of the condition $E_P$ and the latent vector. For the more complex scenario with multiple input conditions, we find that simply concatenating the conditions to the latent vector provides a more compact approach with better overall accuracy. A set of upsampling layers, at the beginning of the generator network, are used to reach the required dimensions before the application of the convolutions [44]. The initial prototype [9] consisted of alternating upsampling layers and convolutional layers similar to ACGAN [45]. By moving the upsampling layers before convolutions we have improved the generator learning process, obtaining a more realistic output. The improvement is most significant for the simulation of cells with very small energy deposits, occurring mostly in the peripheral regions of the image as presented in Figure 2 left panel. The 3DGAN has a stronger generator (seven convolutional layers) than the discriminator (with four layers) in order to cope with the increase in image complexity. A gradually decreasing kernel is used for the generator (to generate more local features in higher-order layers). The geometry of the energy shower, extending along the length of the longitudinal dimension, while being narrow for the transverse dimensions, influenced the choice of the generator kernels in the $Z$ dimension, which are kept larger. Experimentation with kernels did not have any significant effect on model performance thus number of filters and kernel sizes are further adjusted in consideration to memory constraints and image dimensions.

Batch normalization [46] (with a smaller fuzz factor of $10^{-6}$) is applied to all except the first convolutional layer in the discriminator and the last two layers in the generator, these exclusions are aimed at encouraging the large dynamic range for the pixel intensities. Relu [47] activation function is used for the generator layers to induce sparsity while the

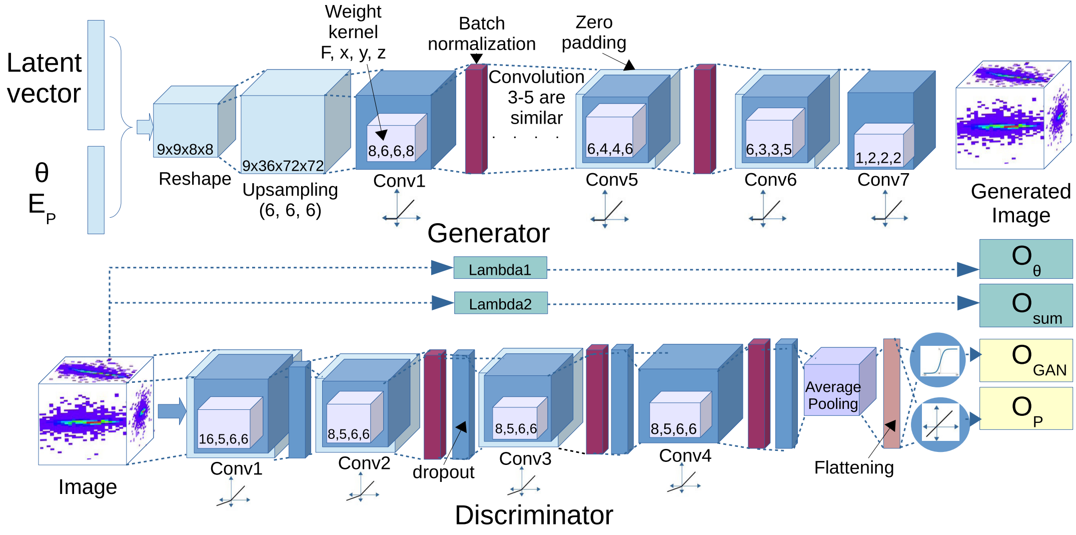


**FIGURE 3.** The 3DGAN architecture, see the text for details.

leakyRelu [48] is used for the discriminator hidden layers. The discriminator is regularized by a dropout [49] of 20% and a single average pooling layer after the last convolutional layer (additional pooling layers result in substantial loss of performance).

The discriminator network has two trainable outputs: $O_{GAN}$ estimates the GAN real/fake probability via a sigmoid neuron and $O_P$ predicts $E_P$ as an auxiliary regression task through a linear neuron. Two additional outputs are implemented as analytical functions of the input image, accomplished through lambda layers representing non-trainable physics-based constraints. The $O_{sum}$ is the total deposited energy ($E_{sum} = \sum I_{pixel}$) and $O_\theta$ is the measured incident angle calculated as a weighted mean of measured angles exploiting the slopes of the lines joining the barycenter of the energy deposition of the event, to the barycenters for the $XY$ planes at each position along the $Z$ axis (weighted by the position along the $Z$ axis). Another constraint enforcing pixel intensities distribution was included in our previous contribution [10], but further tests indicated that constraint to be ineffective, and thus our final model is simplified by removing the redundant output. The architecture is implemented using Keras 2.2.4 [50] (with Tensorflow 1.14.0 [51] as a backend).[2]

### C. LOSS FUNCTION

The 3DGAN loss function is built from the weighted sum of individual terms pertaining to the discriminator outputs and constraints. The introduction of domain-related constraints is essential to achieve a high level of accuracy. Figure 4 shows the effect of constraining the $E_{sum}$. Without the constraint, $E_{sum}$ for generated images has a roughly uniform distribution, but only after applying the constraint, the sum is correctly mapped to $E_P$. It should be noted that by constraining the total deposited energy we make sure that the energy conservation is preserved.

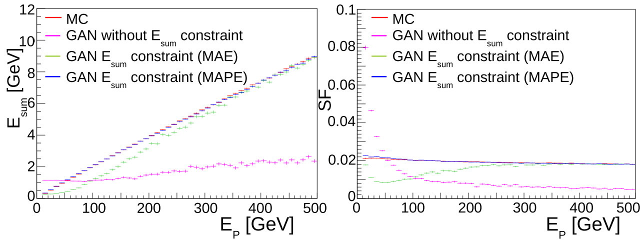


**FIGURE 4.** The GAN performance without $E_{sum}$ constraint (magenta) and with the constraint (green and blue), demonstrated by a comparison between Monte Carlo and GAN generated events for $E_{sum}$ vs. $E_P$ (left) and $SF$ vs. $E_P$ (right). (initial prototype [9]).

Equation (1) shows the different components of the discriminator loss: the discriminator real/fake probability as defined in [13] ($L_G$), the primary particle energy regression ($L_P$), the total deposited energy ($L_E$), and incident angle measurement ($L_A$). The losses are balanced by the corresponding weights $W$. The $L_P$ and $L_\theta$ provide feedback on the conditioning of the image, while $L_E$ and $L_A$ help to impose external constraints on the images. The generator loss is implemented as the inverse of $L_G$ together with the auxiliary losses and constraints.

$$L_{3DGAN} = W_G L_G + W_P L_P + W_A L_A + W_E L_E \qquad (1)$$

The loss components presented in Equation (1) are based on different errors. $L_G$ is implemented as binary cross-entropy. $L_P$ and $L_E$ are evaluated as the mean absolute

[2] The code is available at https://github.com/svalleco/3Dgan

percentage error (MAPE), while $L_A$ is evaluated as the mean absolute error (MAE). Figure 4 also demonstrates how the choice of the error for the loss term affects the quality of results. As $O_{sum}$ and $O_P$ are both correlated to pixel intensities, thus a percentage error results in better performance especially at the lower end of the spectrum where smaller energies are deposited. Apart from some degradation in both simulating and predicting these low intensities, a slight energy difference at the lower end would result in a tiny absolute error but a larger percentage error (thus a larger gradient), therefore improving the training for this region.

The loss weights are selected roughly following the guidelines by Chollet [52] according to the type of losses, probable values at convergence, and relative importance. The losses based on mean percentage error are assigned a weight of 0.1. The $W_A$ is assigned a value of 25 and the BCE losses are assigned a weight of 3. A high level of accuracy is required for the correct $E_{sum}$ of the generated images. Increasing the weight $W_E$ fails to improve the performance on $L_E$ as shown in Figure 5 (right), while the performance on $L_G$ deteriorates (left). Therefore we add another step to select the final network based on the minimum mean absolute relative error on the sampling fraction (using additional holdout samples), in the last ten epochs so as to retain a high level of accuracy on this quantity.

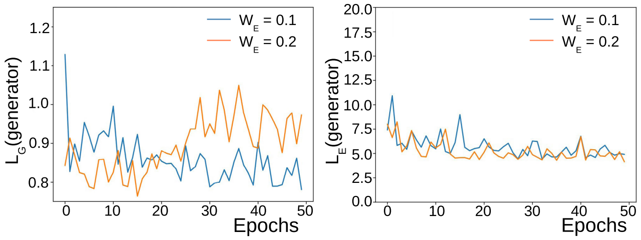


**FIGURE 5.** **The 3DGAN generator test losses for loss weight $W_E$ equal to 0.1 (cyan) and 0.2 (orange). (Left) $L_G$ vs. epochs. (Right) $L_E$ vs. epochs. (initial prototype [9]).**

### D. TRAINING

The GAN training converges when the discriminator cannot discriminate between real and fake samples. The discriminator estimates a real/fake probability close to 50% for both the real and fake images. The convergence for the simulation of the more complex scenario involving both $E_P$ and $\theta$ could not be accomplished directly, only restricting the $E_P$ range could finally allow the training to converge. We thus apply a two-step training based on the transfer learning to simulate the full $E_P$ range. During the first step, we train 3DGAN on events with $E_P$ in the 100 GeV to 200 GeV range. The second training step extends the data to events having $E_P$ from the full 2 GeV to 500 GeV range. The final network is further selected based on the minimum mean relative error (see Section V-F2) on $SF$ in the last ten epochs, evaluated on additional holdout samples.

For each iteration, we train the discriminator on a batch of real images and a batch of generated images (applying label switching) [53]. Adopting a balanced approach, we also train the generator twice while freezing the discriminator weights. Another modification applied to the GAN training process involves generating the fake image batch for the same input conditions as the real image batch instead of randomly sampling the input conditions. Thus not only alleviating the need to find the complex mapping of $E_P$ to $E_{sum}$, as well as having to apply it at run time (as performed in [9]). A speedup of more than 30% could be achieved for the processing time per epoch, as well as removing any chances of error due to inaccurate mapping. A number of optimizers and learning rates were investigated. The RMSProp [54] optimizer with a learning rate of 0.001 was finally selected on the basis of performance, to train the network through Stochastic Gradient Descent in mini-batches of 64 events since larger batch sizes could not be supported due to memory constraints.

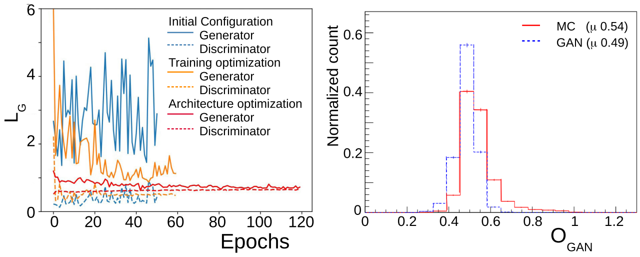


**FIGURE 6.** **The 3DGAN test losses. (Left) $L_G$ loss for the generator (continuous line) and the discriminator (dashed line): applying the configuration of the initial prototype [9] to the variable angle data does not work (teal); pre-processing for the reduction of the pixel intensity dynamic range and additional loss terms improved convergence (orange); architecture optimization (with upsampling layers at the beginning and more convolutional layers further improved the results (red). See text for additional details. (Right) The real/fake probability ($O_{GAN}$) for both the Monte Carlo (red) and GAN generated events (blue dotted) is close to 0.5.**

Figure 6 right panel compares the $L_G$ test losses at different stages of the current work for the restricted $E_P$ range. Restricting the $E_P$ range started showing some improvement in the training losses for the initial configuration [9] but the test losses (blue) remain highly random. Implementing the pre-processing step as explained in section IV-A, adjusting the losses, and generating images for the same inputs as real images denote the training optimization (orange), that allows the losses to start converging. The architecture optimization involves upsampling before convolutions and additional layers in the generator (red), resulting in decreasing the loss further and improving the convergence (the GAN could still not converge for the full $E_P$ range directly thus necessitating the two-step training). The left panel shows how the real/fake probability ($O_{GAN}$) estimated by the discriminator for the real (red) and the fake (blue) images, has very similar distributions, at the end of training.

For the restricted energy range (the first training step) the training runs in 2 hours per epoch on a single NVIDIA GeForce GTX 1080 card for 130 epochs. The second training step using an increased number of events from the whole energy range from 2 GeV to 500 GeV (every epoch lasts four hours), is run for 30 epochs. The training time is limited due to time constraints although over-fitting is not observed even training far beyond these limits, while very little improvement is observed.

## V. RESULTS AND DISCUSSION

As mentioned in Section I, the 3DGAN results cannot be compared directly to previous approaches as the problem is more complex. We thus validate the 3DGAN performance by a detailed comparison to the Monte Carlo simulation. Established fast simulation approaches report an accuracy within 10% of the Monte Carlo, thus we aim to achieve a similar level for the GAN generated events. We test the distributions of several physics quantities as a function of the main inputs to the simulation process ($E_P$ and $\theta$). In order to do so, we select events in 5 GeV bins for $E_P$ and 0.1 rad (5.73°) bins for $\theta$, as well as some unbinned events, from the unseen data of 100, 000 random events. The comparisons are performed with GAN events generated for the same $E_P$ and $\theta$ values as the MC events. We have computed the mean absolute relative errors for the histogram bins (BMRE). The detailed physics validation involves numerous feature distributions compared for different input bins, thus resulting in hundreds of histograms. The GAN generated events have a close agreement with the Monte Carlo, greatly surpassing our aimed accuracy level for most of the physics features. Only a few representative features are presented here in order to simplify the discussion.

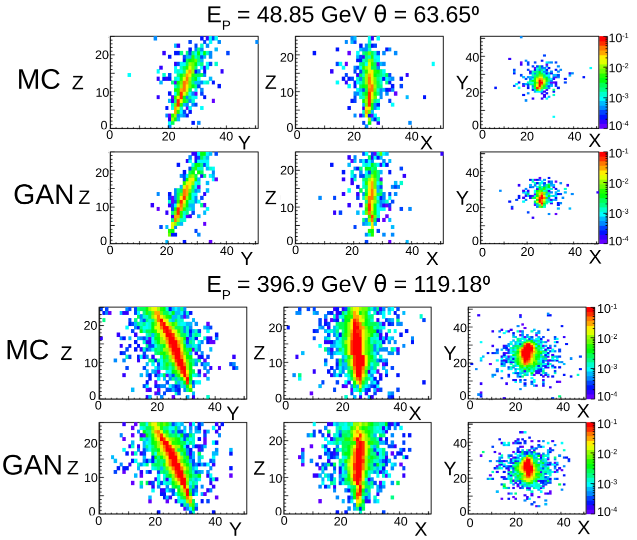


**FIGURE 7.** Example 2D shower sections [log] on the *YZ*, *XZ*, and *XY* planes with Ep and $\theta$ sampled from both ends of their respective spectra. The GAN events are generated for the same $E_P$ and $\theta$ values as the MC events. $E_P$ = 48.85 GeV and $\theta$ = 63.65° (top two rows). $E_P$ = 396.9 GeV and $\theta$ = 119.18° (bottom two rows).

### A. VISUAL COMPARISON

In order to perform a preliminary visual inspection of the results, we build 2D projections on different planes of three dimensional events. Figure 7 presents an example of 2D sections of the MC and GAN showers corresponding to electrons entering the calorimeter with different energies and angles (selected from the tails of the energy and angle distributions). The images appear visually similar, sharp, and unique.

### B. SAMPLING FRACTION AND CALORIMETER RESPONSE

As explained earlier the Sampling Fraction (*SF*), is correlated to the amount of energy recorded by the calorimeter and it depends on the particular detector geometry and the primary particle type and energy. Figure 8 demonstrates the close agreement over the entire energy range for both ends of the $\theta$ spectrum. It should be noted here that although for the 62° bin there is a slightly lower *SF* for the GAN images, yet the difference is very small with a mean absolute relative error of a few percent for the histogram bins.

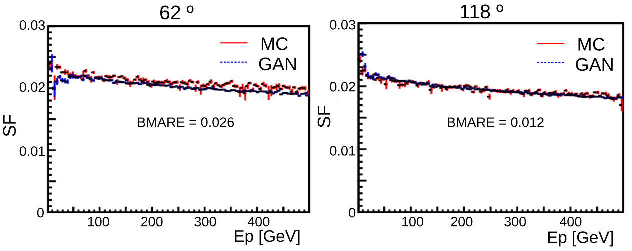


**FIGURE 8.** The sampling fraction for Monte Carlo vs. GAN events with $E_P$ from 2 GeV to 500 GeV. The $\theta$ values are from 62° bin (left) and 118° bin (right).

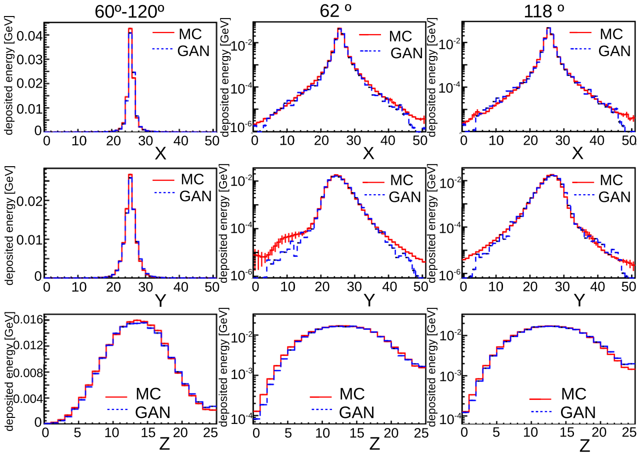


**FIGURE 9.** Shower Shapes for Monte Carlo vs. GAN events along *X*, *Y*, and *Z* axis with $E_P$ range of 2 GeV to 500 GeV. The shapes for events with $\theta$ range of 60° to 120° are plotted on linear y-scale (left column), and bins of 62° and 118° on logarithmic y-scales (two columns on the right).

### C. PARTICLE SHOWER SHAPES

Figure 9 presents the energy profiles for the showers along the $X$, $Y$ and $Z$ axes, both in linear (full $\theta$ range of 60° to 120°) and log scale (62° and 118° bins of $\theta$). The network is capable of correctly reproducing the spatial distribution of energy deposits as a function of the incident

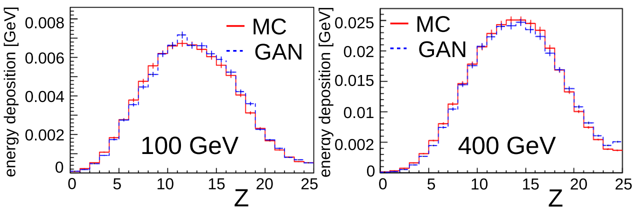


**FIGURE 10.** **Shower shapes along the $Z$ axis for Monte Carlo vs. GAN events in different primary energy bins. (Left) $E_P = 100$ GeV. (Right) $E_P = 400$ GeV.**

angle, across a large dynamic range. In the log scale, some discrepancies are observed at the edges of the simulated volumes, where smaller energy depositions occur. The amount of energy expected in this region (well below $10^{-4}$ GeV) is below the threshold for this detector and comparable to the pedestal values. Figure 10 presents the longitudinal projection ($XY$) for the MC and GAN showers for some $E_P$ bins proving that the agreement in shapes is maintained along the entire spectrum.

## D. SPARSITY AND CELL ENERGY DISTRIBUTION

The detector cells are mostly empty with energy deposited only in less than 20% of them. Figure 11 (right) shows the fraction of cells where some energy is deposited as a function of the threshold used for the cutoff. It can be seen that the Monte Carlo and GAN images present a similar level of sparsity. The energies deposited in the calorimeter cells are our pixels intensities. Figure 11 (left) shows the MC and the GAN agreement in terms of pixel intensities down to very low values. It should be noted that at around $0.2 \times 10^{-3}$ GeV the MC intensities show a sharp, vertical drop. As expected the GAN smooths out this cut.

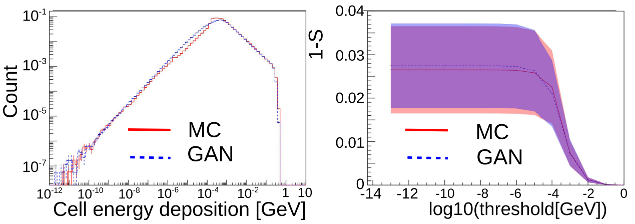


**FIGURE 11.** **Monte Carlo vs. GAN shower features for 2 GeV to 500 GeV electrons show close agreement. (Left) Cell energy distributions. (Right) Sparsity as the fraction of cells above a threshold for different values of threshold.**

## E. CORRELATIONS

We study the internal correlation between shower features as well as the correlation between MC and GAN images. The correlations among different shower features and the inputs i.e. energy and angle should be preserved. We evaluate the correlation matrix calculated on different quantities such as the shower shapes, $E_{sum}$, $E_P$, $|90° - \theta|$, and numbers of hits above a threshold ($0.2 \times 10^{-3}$ GeV). Figure 12, the left panel, presents the difference between the internal correlations present in the MC data and those of the GAN images. These correlations agree to a considerable extent with a mean error of less than 10%. The incident angle $\theta$ measured in MC and GAN, shown in Figure 12, right panel, also presents a high level of agreement.

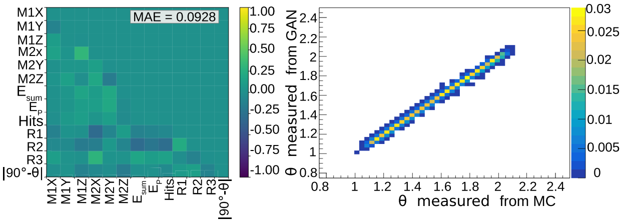


**FIGURE 12.** **(Left) The difference between correlation matrices exhibiting internal correlations among shower inputs and physics features for Monte Carlo and GAN generated events. (Right) Correlation between $\theta$ measured from Monte Carlo events and that measured from GAN events generated for similar $\theta$ values.**

## F. FURTHER VALIDATION

The main validation of 3DGAN performance involved detailed comparison of physics-based features across the whole range of the primary energies and the incident angles as explained in Section III. We have further explored validation from diverse standpoints including further mathematical formulation of the physics performance, as well as validation by a neural network and image quality metrics.

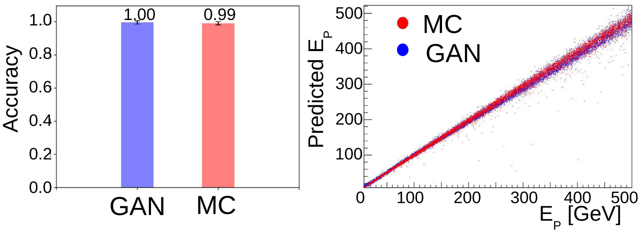


**FIGURE 13.** **Triforce results for GAN vs. Monte Carlo events (9834 electron events for each type). (Left) Classification accuracy. (Right) Scatter plot between true and predicted $E_P$.**

### 1) VALIDATION BY THIRD PARTY NETWORK

The most popular metrics used in GAN literature involve an external pre-trained network. Similar to what is done in general image generation problems, where the output of the inception network is used to quantify the generated image quality [55], [56], we have used the independently developed pre-trained (trained on the same dataset) classifier and regressor network from Triforce [57] to validate our results. We stress here that the actual accuracy is not relevant for our test as that depends on the Triforce performance, but obtaining a similar performance level for GAN and MC events is. Figure 13 shows the classification and regression results obtained by running Triforce on 9834 images from both GAN and MC test samples. The GAN images show a similar response from the Triforce network, as MC images.

Thus proving that the GAN images contain most of the features of the MC data, as learnt by the Triforce network.

### 2) QUANTIFYING PHYSICS PERFORMANCE

The physics performance validated in Section V is based on the study of binned distributions, therefore we test several histogram distances. The metrics like Chi2 and Kolmogorov get saturated for small discrepancies in a few bins. These metrics are highly dependent on external factors like binning and ranges and thus are not suitable for combining as averages. On the other hand, mean errors do not describe well the performance over the whole distribution. In order to further quantify the performance, we calculate the relative errors for individual bins of a distribution. The mean of all bin errors is then treated as a metric. We have calculated this metric on physics features like the shower shapes, moments, and sampling fraction. These metrics can also be combined as a single figure for a future hyper-parameter effort.

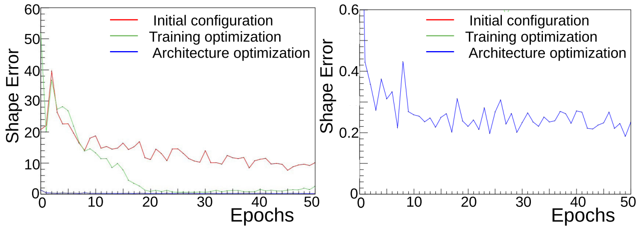


**FIGURE 14. Mean relative errors on shower shapes for the first training step with restricted $E_P$ range. (Left) Initial implementation without optimization (red), with pre-processing and additional losses (green), and after applying architecture modifications (blue). (Right) Reducing the $Y$ axis scale to observe the error for the 3DGAN final version (blue), where it can be seen that the error is now saturated for further analysis.**

In the current work we will only explore the metric as a function of training epochs for the different stages of 3DGAN development. The shower shapes provide an example of geometrical features that are learned by the model implicitly. Figure 14 presents the error for shower shapes associated with different versions of 3DGAN as described in Section IV-D for the first step of training (with restricted $E_P$ range). The initial configuration has high errors with mean and variance decreasing with epochs (red). The high values of error can be attributed to the tails of the distributions, where the energy deposition is very low. The training optimization greatly reduces the error yet a slight over-fitting can be detected in the later epochs (green). In the final implementation (blue) the shape distributions improve significantly. As observed from Figure 14 right panel, the training has reached a level where the error remains low yet not decreasing much with the epochs and the MRE metric seems to be saturated.

The sampling fraction is an example of analytical functions of the image. The simulation requires a very precise $SF$ and its small variation through epochs is significant for performance. Figure 15 right panel presents the error on $SF$ averaged over 5 epochs. It can be observed that the error is below 10% for most of the training but in order to further improve the accuracy, we select the final network among the last few epochs (10 for current application) based on the minimum mean relative error, evaluated on holdout samples.

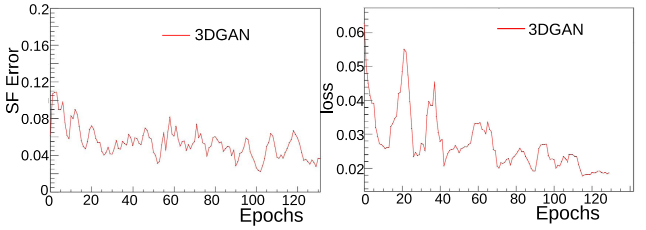


**FIGURE 15. Validation of the 3DGAN final version for the restricted energy range training step. (Left) Mean relative error for $SF$ vs. epochs (with average smoothing of order 5). (Right) Entropic Gromove Wasserstein Discrepancy calculated on correlation matrix vs. epochs (with average smoothing of order 5).**

The Wasserstein Discrepancy [58] is the next figure of merit we have tested. The Wasserstein distance [59] is an optimal transport distance metric that can be used for comparison of distributions. Wasserstein Discrepancy is measured on "similarity matrices" from two sets of samples. We have computed the Wasserstein Discrepancy from the correlation matrix (Figure 12). Figure 15, the right panel, presents the metric (with average smoothing) as the training progresses through epochs. There is a steady decrease in both mean and variance (even after mean relative errors presented in Figure 14 cease to improve).

### 3) IMAGE QUALITY ANALYSIS

The assessment of the similarity/diversity of the generated sample compared to the original sample is an important step for the development of generative models. In the case of simulation, the shower produced by particles with the same $E_P$ and $\theta$ should show similar features while retaining the expected statistical variance. In order to evaluate this aspect, we utilize some metrics for image quality assessment.

The Structural Similarity Index Measure (SSIM) [60] has been used for GAN performance evaluation [34]. The SSIM quantifies the quality of images based on their similarity to a reference. For images x and y, the SSIM is computed between windows from both x and y. If the mean and standard deviation of both windows are $\mu_x$, $\sigma_x$ and $\mu_y$, $\sigma_y$ respectively, then SSIM is given by the following equation (2):

$$\mathrm{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{2}$$

where $C_1 = (k_1\, L)^2$ and $C_2 = (k_2\, L)^2$ with $k_1 = 0.01$ and $k_2 = 0.03$. The value of "L" is usually taken to be equal to the dynamic range of the pixel intensities. The parameter $L$ thus determines the fuzz factor added against the weak denominator to avoid zero division. A value of SSIM close to one indicates very similar images while a lower value indicates more diverse images. In the context of GAN, the similarity metric is also treated as a measure of diversity.

The SSIM is sensitive to the pixel dynamic range (by the virtue of parameter $L$) and should be adjusted accordingly. Figure 11 (left) shows that our ''pixel intensities'' have a very different dynamic range as compared to the standard RGB case, with a logarithmic difference between the maximum and minimum values. Therefore, the experiment is repeated for different values of $L$ in order to verify the sensitivity of the SSIM to our range of pixel intensities (energy depositions).

We measure SSIM for MC images against MC images, MC images against GAN images, and GAN images against GAN images. The events in each set are random events from the same $E_P$ and $\theta$ bin. The SSIM value from MC vs. MC images can act as a reference to assess the GAN performance.

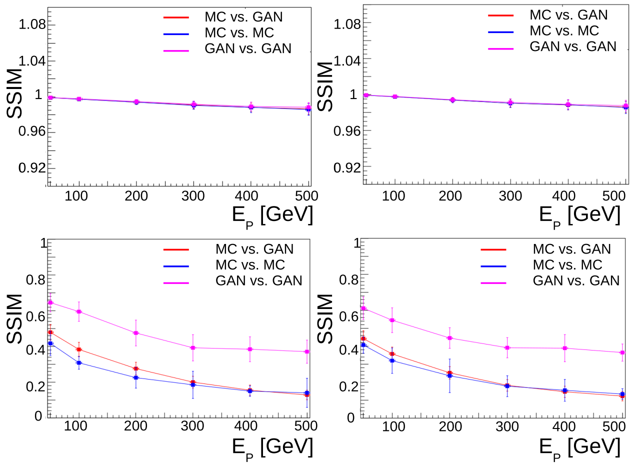


**FIGURE 16. SSIM for $L = 1$ (top row) and $L = 0.0001$ (bottom row). The SSIM is computed for MC vs. GAN events (red), MC vs. MC events (blue), and GAN vs. GAN events (magenta) as a function of $E_P$. (Left column) $\theta = 62°$ bin. (Right column) $\theta = 118°$ bin.**

The SSIM value for $L = 1$ (as generally suggested for float intensities) is close to one and identical for all three sets of images as shown in Figure 16 top row, indicating that the index is not sensitive to the difference present between individual samples in a bin at this scale. The SSIM values decrease with $L$ and a difference can be observed between MC vs. MC and GAN vs. GAN (indicating that the metric is now more sensitive). Decreasing $L$ below $10^{-4}$, has no further effect. Figure 16 bottom row shows the SSIM for $L$ equal to $10^{-4}$. The GAN vs. GAN SSIM is slightly higher than MC vs. MC, indicating that the GAN samples have less diversity in a bin, but is of the same order. The SSIM for MC vs. GAN is similar to MC vs. MC, proving that GAN images come from the same distribution as MC images. The figure also shows that SSIM is constant across the $\theta$ spectrum.

The Mean Subtracted Contrast Normalized Coefficients (MSCN Coefficients) [61] have been used for blind image quality assessment. The MSCN Coefficients are computed by taking the mean and the standard deviation of windows in an image. Let $\mu$ and $\sigma$ be the mean and standard deviation of an image window then a pixel intensity $I$ will be converted to coefficient $I^*$ by the following equation:

$$I^* = \frac{(I - \mu)}{\sigma} \tag{3}$$

These MSCN Coefficients can then be mapped to an image quality score [61]. The distributions of the MSCN Coefficients from real and fake images have also been used for GAN evaluation [62]. The 3DGAN images are highly sparse with small energy depositions thus the Equation (3) is modified as following :

$$I^* = \frac{(I - \mu_{nonzero})}{\sigma_{nonzero} + epsilon} \tag{4}$$

Here we stabilize by adding a fuzz factor in the denominator (taking epsilon value equal to $10^{-7}$). The mean ($\mu_{nonzero}$) and the standard deviation ($\sigma_{nonzero}$) are evaluated considering only the nonzero entries. Figure 17 compares the histograms of the coefficients as computed using Equation (4) for MC and GAN generated images. The nature of our data is very different from natural scenes and thus the distributions are not Gaussian. The MSCN Coefficient distributions can still be treated as statistical signatures and are almost identical for the real and fake images, showing very similar features. The Jensen Shannon Distance ($JSD$) between the real and fake distributions is very small, thus indicating the histograms to be highly similar.

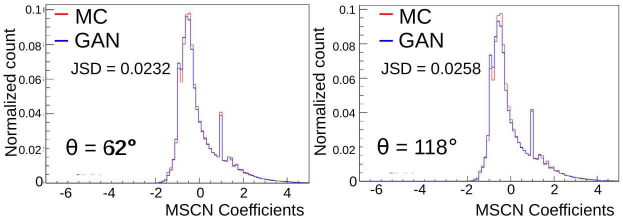


**FIGURE 17. MSCN coefficient histograms for MC and GAN images for $E_P = 2 - 500$ GeV. Left) $\theta = 62°$. Right) $\theta = 118°$. The JSD denotes the Jensen Shannon Distance between the two distributions.**

## VI. CONCLUSION

The Monte Carlo simulation of detector response is a time and resource intensive task. There is a drive in the HEP community to find possible fast alternatives while maintaining physics accuracy. Fastsim is a set of established techniques usually replacing parts of the detailed simulation where some loss in accuracy can be traded for speed. These techniques have been incorporated into existing particle simulation packages frequently employed in practical applications such as GFlash [5], Altfast [4], and FastCaloSim [63] etc. We demonstrate that the fast simulation based on deep learning can surpass these methodologies both in speed and accuracy. The detector response can be generated as images and easily integrated into a detailed simulation, in the same manner as existing fast simulation methodologies.

The 3DGAN model is capable of reproducing single particle (electron) showers for a high granularity calorimeter representative of detectors designed for future particle accelerators. We have validated the 3DGAN performance by comparing it with the classical Monte Carlo in great detail. The agreement is within a few percent over a very large dynamic range. This impressive level of agreement surpasses similar deep learning-based applications to HEP simulation [30] and fast simulation approaches based on classical methods [5]. The inference time on Intel Xeon 8180 is about 13.4 m sec per particle whereas about 3.5 m sec per particle on GeForce GTX 1080. For comparison, GEANT4 can simulate a similar shower in about 17 seconds per particle on an Intel Xeon 8180 (currently it is not possible to run a full GEANT4-based simulation on GPUs). Therefore, the 3DGAN provides three orders of magnitude speedup.

The main R&D directions we are following are the investigation into the generation of rare modes, the generalization of the model for different particles and detectors as well as the inclusion of other detector information. Methods such as ensembling [64] or mode regularization [65] techniques can be employed to improve the performance for rare modes present in the data. The 3DGAN approach for calorimeter simulation has been applied to the electron showers and the same methodology can be extended to other types of particles. The simulation of some particles might require the inclusion of further detector information. The current fastsim approaches are specific to a particular type of detector due to the great variation present in the design of different detectors. The detector geometry defines the overall detector layout, material, cell size, and consequently image dimensions and features. All of these factors contribute to the characteristics of the data and thus require specific optimization of the architecture and the training procedure. The GAN technique can be exploited for simulation of any detector output as an image but the current methodologies require individual adaptation and development for best performance. Some preliminary efforts towards future generalization in terms of detector geometry have been undertaken. A hyper-parameter scan can facilitate automatic tuning at the cost of extremely long training times (not currently feasible). A distributed computing approach [66], [67] was explored to decrease the training time for future development towards hyper-parameter optimization. A preliminary investigation was also conducted to explore the Genetic Algorithm that can optimize both weights and architecture at the same time [67].

## ACKNOWLEDGMENT

This work has been conducted with the support of Intel in the framework of the CERN openlab-Intel collaboration agreement. Part of this work was conducted at ''Banks,'' the AI GPU cluster at Caltech. The authors acknowledge NVIDIA, SuperMicro, and the Kavli Foundation for their support of ''iBanks.'' They thank Matt Zhang from the University of Illinois at Urbana-Champaign for helping in validation by the Triforce model.

## REFERENCES

[1] *Worldwide LHC Computing Grid*. Accessed: Sep. 26, 2021. [Online]. Available: http://wlcg.web.cern.ch/

[2] J. Albrecht *et al.*, ''A roadmap for HEP software and computing R&D for the 2020s,'' *Comput. Softw. Big Sci.*, vol. 3, no. 1, p. 7, Mar. 2019.

[3] G. Apollinari, I. A. Béjar, O. Brüning, P. Fessia, M. Lamont, L. Rossi, and L. Tavian, ''High-luminosity large hadron collider (HL-LHC): Technical design report V. 0.1,'' CERN, Geneva, Switzerland, Tech. Rep. CERN-2017-007-M, 2017, pp. 1–516, vol. 4, doi: 10.23731/CYRM-2017-004.

[4] W. Lucas, ''Fast simulation for ATLAS: Atlfast-II and ISF,'' in *Proc. Int. Conf. Comput. High Energy Nucl. Phys.*, vol. 396, 2012, pp. 1–10.

[5] D. Orbaker, ''Fast simulation of the CMS detector,'' in *Proc. Int. Conf. Comput. High Energy Nucl. Phys.*, vol. 219, 2010, pp. 1–7.

[6] D. Autiero *et al.*, ''Parameterization of $e$ and $\gamma$ initiated showers in the NOMAD lead-glass calorimeter,'' *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip.*, vol. 425, p. 188, Aug. 1998.

[7] E. Barberio, J. Boudreau, B. Butler, S. L. Cheung, A. Dell'Acqua, A. D. Simone, E. Ehrenfeld, M. V. Gallas, A. Glazov, Z. Marshall, J. Mueller, R. Plačakyte, A. Rimoldi, P. Savard, V. Tsulaia, A. Waugh, and C. C. Young, ''Fast simulation of electromagnetic showers in the ATLAS calorimeter: Frozen showers,'' *J. Phys., Conf. Ser.*, vol. 160, Apr. 2009, Art. no. 012082.

[8] J. Schaarschmidt *et al.*, ''The new fast calorimeter simulation in ATLAS,'' ATLAS Collaboration, CERN, Geneva, Switzerland, Tech. Rep. ATL-SOFT-PUB-2018-002, Jul. 2018. [Online]. Available: https://cds.cern.ch/record/2630434

[9] G. R. Khattak, S. Vallecorsa, and F. Carminati, ''Three dimensional energy parametrized generative adversarial networks for electromagnetic shower simulation,'' in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 3913–3917.

[10] G. R. Khattak, S. Vallecorsa, F. Carminati, and G. M. Khan, ''Particle detector simulation using generative adversarial networks with domain related constraints,'' in *Proc. 18th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2019, pp. 28–33.

[11] Y. Bengio, E. Thibodeau-Laufer, and J. Yosinski, ''Deep generative stochastic networks trainable by backprop,'' *CoRR*, vol. abs/1306.1091, pp. II-226–II-234, 2013. [Online]. Available: http://arxiv.org/abs/1306.1091 and http://dblp.uni-trier.de/rec/bib/journals/corr/BengioT13

[12] D. P Kingma and M. Welling, ''Auto-encoding variational Bayes,'' Dec. 2013, *arXiv:1312.6114*. [Online]. Available: http://arxiv.org/abs/1312.6114

[13] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, ''Generative adversarial nets,'' in *Proc. 27th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, vol. 2. Cambridge, MA, USA: MIT Press, 2014, pp. 2672–2680.

[14] D. Fudenberg and J. Tirole, *Game Theory*. Cambridge, MA, USA: MIT Press, 1991.

[15] M. Arjovsky, S. Chintala, and L. Bottou, ''Wasserstein GAN,'' 2017, *arXiv:1701.07875*. [Online]. Available: https://arxiv.org/abs/1701.07875

[16] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, ''StackGAN++: Realistic image synthesis with stacked generative adversarial networks,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1947–1962, Aug. 2019.

[17] T. Karras, T. Aila, S. Laine, and J. Lehtinen, ''Progressive growing of GANs for improved quality, stability, and variation,'' 2017, *arXiv:1710.10196*. [Online]. Available: http://arxiv.org/abs/1710.10196

[18] S. Li, S. Jang, and Y. Sung, ''Automatic melody composition using enhanced GAN,'' *Mathematics*, vol. 7, no. 10, p. 883, Sep. 2019.

[19] S. Subramanian, S. Rajeswar, F. Dutil, C. Pal, and A. Courville, ''Adversarial generation of natural language,'' in *Proc. 2nd Workshop Represent. Learn. NLP*. Vancouver, BC, Canada: Association for Computational Linguistics, Aug. 2017, pp. 241–251.

[20] X. Yi, E. Walia, and P. Babyn, ''Generative adversarial network in medical imaging: A review,'' *Med. Image Anal.*, vol. 58, Dec. 2019, Art. no. 101552.

[21] Z. Chen, Z. Zeng, H. Shen, X. Zheng, P. Dai, and P. Ouyang, ''DN-GAN: Denoising generative adversarial networks for speckle noise reduction in optical coherence tomography images,'' *Biomed. Signal Process. Control*, vol. 55, Jan. 2020, Art. no. 101632.

[22] A. Alsaiari, R. Rustagi, A. Alhakamy, M. M. Thomas, and A. G. Forbes, ''Image denoising using a generative adversarial network,'' in *Proc. IEEE 2nd Int. Conf. Inf. Comput. Technol. (ICICT)*, Mar. 2019, pp. 126–132.

[23] J. Kim, K. Jeong, H. Choi, and K. Seo, ''GAN-based anomaly detection in imbalance problems,'' in *Computer Vision—ECCV 2020 Workshops*, A. Bartoli and A. Fusiello, Eds. Cham, Switzerland: Springer, 2020, pp. 128–145.

[24] C. Han, L. Rundo, K. Murao, T. Noguchi, Y. Shimahara, Z. Á. Milacski, S. Koshino, E. Sala, H. Nakayama, and S. Satoh, "MADGAN: Unsupervised medical anomaly detection GAN using multiple adjacent brain MRI slice reconstruction," *BMC Bioinf.*, vol. 22, no. S2, p. 31, Apr. 2021.

[25] R. M. Feder, P. Berger, and G. Stein, "Nonlinear 3D cosmic web simulation with heavy-tailed generative adversarial networks," *Phys. Rev. D, Part. Fields*, vol. 102, no. 10, Nov. 2020, Art. no. 103504.

[26] N. Perraudin, S. Marcon, A. Lucchi, and T. Kacprzak, "Emulation of cosmological mass maps with conditional generative adversarial networks," *Frontiers Artif. Intell.*, vol. 4, Jun. 2021, Art. no. 673062.

[27] R. Rong, S. Jiang, L. Xu, G. Xiao, Y. Xie, D. J. Liu, Q. Li, and X. Zhan, "MB-GAN: Microbiome simulation via generative adversarial network," *GigaScience*, vol. 10, no. 2, pp. 1–15, Feb. 2021.

[28] Y. Dan, Y. Zhao, X. Li, S. Li, M. Hu, and J. Hu, "Generative adversarial networks (GAN) based efficient sampling of chemical composition space for inverse design of inorganic materials," *npj Comput. Mater.*, vol. 6, no. 1, p. 84, Jun. 2020.

[29] L. de Oliveira, M. Paganini, and B. Nachman, "Learning particle physics by example: Location-aware generative adversarial networks for physics synthesis," 2017, *arXiv:1701.05927*. [Online]. Available: http://arxiv.org/abs/1701.05927

[30] M. Paganini, L. de Oliveira, and B. Nachman, "CaloGAN: Simulating 3D high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks," *Phys. Rev. D, Part. Fields*, vol. 97, no. 1, Jan. 2018, Art. no. 014021.

[31] D. Salamani, S. Gadatsch, T. Golling, G. A. Stewart, A. Ghosh, D. Rousseau, A. Hasib, and J. Schaarschmidt, "Deep generative models for fast shower simulation in ATLAS," in *Proc. IEEE 14th Int. Conf. e-Sci. (e-Science)*, Oct./Nov. 2018, p. 348.

[32] R. Di Sipio, M. F. Giannelli, S. K. Haghighat, and S. Palazzo, "DijetGAN: A generative-adversarial network approach for the simulation of QCD dijet events at the LHC," *J. High Energy Phys.*, vol. 2019, no. 8, p. 110, Aug. 2019.

[33] V. Chekalina, E. Orlova, F. Ratnikov, D. Ulyanov, A. Ustyuzhanin, and E. Zakharov, "Generative models for fast calorimeter simulation: The LHCb case," in *Proc. EPJ Web Conf.*, vol. 214, 2019, p. 02034.

[34] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," Oct. 2016, *arXiv:1610.09585*. [Online]. Available: http://arxiv.org/abs/1610.09585

[35] S. Alonso-Monsalve and L. H. Whitehead, "Image-based model parameter optimization using model-assisted generative adversarial networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 12, pp. 5645–5650, Dec. 2020.

[36] E. Buhmann, S. Diefenbacher, E. Eren, F. Gaede, G. Kasieczka, A. Korol, and K. Krüger, "Getting high: High fidelity simulation of high granularity calorimeters with high speed," May 2020, *arXiv:2005.05334*. [Online]. Available: http://arxiv.org/abs/2005.05334

[37] P. Lebrun, L. Linssen, A. Lucaci-Timoce, D. Schulte, F. Simon, S. Stapnes, N. Toge, H. Weerts, and J. Wells, "The CLIC programme: Towards a staged e+e− linear collider exploring the terascale: CLIC conceptual design report," Sep. 2012, *arXiv:1209.2543*. [Online]. Available: http://arxiv.org/abs/1209.2543

[38] *Linear Collider Detector Project at Cern*, CERN, Meyrin, Switzerland, 2014.

[39] CERN. *Welcome to the Compact Linear Collider Website*. Accessed: Jan. 29, 2021. [Online]. Available: http://clic-study.web.cern.ch

[40] S. Agostinelli *et al.*, "GEANT4: A simulation toolkit," *Nucl. Instrum. Meth.*, vol. 506, pp. 250–303, Jul. 2003.

[41] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst. (NIPS)*. Red Hook, NY, USA: Curran Associates, 2016, pp. 2180–2188.

[42] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[43] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proc. 4th Int. Conf. Learn. Represent. (ICLR)*, Y. Bengio and Y. LeCunSan, Eds. Juan, Puerto Rico, May 2016. [Online]. Available: http://arxiv.org/abs/1511.06434 and https://dblp.org/rec/journals/corr/RadfordMC15.bib

[44] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, vol. 1, no. 10, p. e3, Oct. 2016.

[45] *Keras-Team*. Accessed: Sep. 30, 2010. [Online]. Available: https://github.com/keras-team/keras

[46] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, vol. 37, 2015, pp. 448–456.

[47] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, Oct. 2010, pp. 807–814.

[48] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML Workshop Deep Learn. Audio, Speech Lang. Process.*, 2013, pp. 1–6.

[49] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[50] F. Chollet. (2015). Keras. [Online]. Available: https://github.com/fchollet/keras

[51] TensorFlow Developers, "TensorFlow, v2.6.0-rc1," Zenodo, Jul. 2021, doi: 10.5281/zenodo.5095721.

[52] F. Chollet, *Deep Learning With Python*, 1st ed. Shelter Island, NY, USA: Manning Publications, 2017.

[53] S. Chintala, E. Denton, M. Arjovsky, and M. Mathieu, "How to train a GAN? Tips and tricks to make GANs work," GitHub, Dec. 2016. [Online]. Available: https://github.com/soumith/ganhacks

[54] G. Hinton, N. Srivastava, and K. Swersky, "Lecture 6a overview of mini–batch gradi-ent descent," 2012. [Online]. Available: https://www.cs.toronto.edu/~hinton/coursera/lecture6/lec6.pdf

[55] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst. (NIPS)*. Red Hook, NY, USA: Curran Associates, 2016, pp. 2234–2242.

[56] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," Jun. 2017, *arXiv:1706.08500*. [Online]. Available: http://arxiv.org/abs/1706.08500

[57] D. Belayneh, F. Carminati, A. Farbin, B. Hooberman, G. Khattak, M. Liu, J. Liu, D. Olivito, V. B. Pacela, M. Pierini, A. Schwing, M. Spiropulu, S. Vallecorsa, J.-R. Vlimant, W. Wei, and M. Zhang. (2019). *Calorimetry With Deep Learning: Particle Simulation and Reconstruction for Collider Physics*. [Online]. Available: https://inspirehep.net/literature/1770936

[58] G. Peyré, M. Cuturi, and J. Solomon, "Gromov-Wasserstein averaging of kernel and distance matrices," in *Proc. 33rd Int. Conf. Mach. Learn.*, M. F. Balcan and K. Q. Weinberger, Eds., New York, NY, USA, vol. 48, Jun. 2016, pp. 2664–2672.

[59] A. Ramdas, N. Garcia, and M. Cuturi, "On wasserstein two sample testing and related families of nonparametric tests," Sep. 2015, *arXiv:1509.02237*. [Online]. Available: http://arxiv.org/abs/1509.02237

[60] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[61] L. Song, C. Chen, Y. Xu, G. Xue, and Y. Zhou, "Blind image quality assessment based on a new feature of nature scene statistics," in *Proc. IEEE Vis. Commun. Image Process. Conf.*, Dec. 2014, pp. 37–40.

[62] P. Kancharla and S. S. Channappayya, "Improving the visual quality of generative adversarial network (GAN)-generated images using the multiscale structural similarity index," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 3908–3912.

[63] A. Collaboration, "The atlas calorimeter simulation FastCaloSim," in *Proc. IEEE Nucl. Sci. Symp. Med. Imag. Conf.*, Oct. 2010, pp. 1–5.

[64] J. Xie, B. Xu, and C. Zhang, "Horizontal and vertical ensemble with deep representation for classification," *CoRR*, vol. abs/1306.2759, 2013. [Online]. Available: http://arxiv.org/abs/1306.2759 and https://dblp.org/rec/journals/corr/XieXZ13.bib

[65] T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li, "Mode regularized generative adversarial networks," 2016, *arXiv:1612.02136*. [Online]. Available: http://arxiv.org/abs/1612.02136

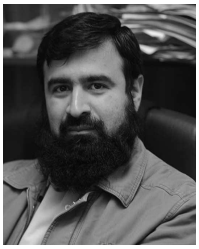
[66] S. Vallecorsa, D. Moise, F. Carminati, and G. R. Khattak, "Data-parallel training of generative adversarial networks on HPC systems for HEP simulations," in *Proc. IEEE 25th Int. Conf. High Perform. Comput. (HiPC)*, Dec. 2018, pp. 162–171.

[67] F. Carminati *et al.*, "Generative adversarial networks for fast simulation: Distributed training and generalisation," in *Proc. Artif. Intell. Sci., Ind. Soc. PoS (AISIS)*, vol. 372, 2020, p. 12, doi: 10.22323/1.372.0012.

**GUL RUKH KHATTAK** received the master's degree in microelectronics from the University of Bolton, U.K., in 2013. She is currently pursuing the Ph.D. degree with the University of Engineering and Technology Peshawar, Peshawar, Pakistan. She is working as a Doctoral Student at CERN openlab, where she investigate deep neural networks for fast detector simulation. She has also worked as a Research Assistant at the Center for Intelligent Systems and Networks Research (CISNR), Peshawar.

**SOFIA VALLECORSA** is currently a CERN Physicist with extensive experience in software development for the HEP domain, in particular on deep learning and quantum computing applications within CERN openlab. Before joining openlab, she has been responsible for the development of deep learning-based technologies for the simulation of particle transport through detectors at CERN and she has worked on optimization of the GeantV detector simulation prototype on modern hardware architectures.

**FEDERICO CARMINATI** received the master's degree in physics from the University of Pavia, Italy, in 1981, and the Ph.D. degree in physics from the University of Nantes, France, in 2013. He is currently the Chief Innovation Officer of CERN openlab, where he has investigating potential applications of machine learning and quantum computing for HEP. Before joining CERN, in 1985, he has worked as a Particle Physicist at Los Alamos National Laboratory and the California Institute of Technology. From 2013 to 2017, he led the development of the new generation of code used to simulate particle transport at CERN.



**GUL MUHAMMAD KHAN** received the Ph.D. degree in electronics engineering from the University of York, U.K., with a specialization in intelligent systems design. He is a prominent figure in AI and the Director of the Center for Intelligent Systems and networks Research (CISNR), Peshawar, Pakistan, and a Faculty Member with the University of Engineering and Technology Peshawar, Peshawar. He is also the Director of the National Center of AI, UET, Peshawar. He has authored *Evolution of Artificial Neural Development: In Search of Learning Genes* (Springer).