Fast Simulation of a High Granularity Calorimeter by Generative Adversarial Networks

Gul Rukh Khattak,[a,b,1] **Sofia Vallecorsa,**[a] **Federico Carminati,**[a] **Gul Muhammad Khan**[b]

[a]*CERN,*
1211 Geneva 23, Switzerland

[b]*University of Engineering and Technology Peshawar,*
Jamrud Road Peshawar Khyber Pakhtunkhwa, Pakistan

E-mail: gul.rukh.khattak@cern.ch, sofia.vallecorsa@cern.ch,
federico.carminati@cern.ch, gk502@uetpeshawar.edu.pk

ABSTRACT: We present the 3DGAN for the simulation of a future high granularity calorimeter output as three-dimensional images. We prove the efficacy of Generative Adversarial Networks (GANs) for generating scientific data while retaining a high level of accuracy for diverse metrics across a large range of input variables. We demonstrate a successful application of the transfer learning concept: we train the network to simulate showers for electrons from a reduced range of primary energies, we then train further for a five times larger range (the model could not train for the larger range directly). The same concept is extended to generate showers for other particles (photons and neutral pions) depositing most of their energies in electromagnetic interactions. In addition, the generation of charged pion showers is also explored, a more accurate effort would require additional data from other detectors not included in the scope of the current work. Our further contribution is a demonstration of using GAN-generated data for a practical application. We train a third-party network using GAN-generated data and prove that the response is similar to a network trained with data from the Monte Carlo simulation.

The showers generated by GAN present accuracy within 10% of Monte Carlo for a diverse range of physics features, with three orders of magnitude speedup. The speedup for both the training and inference can be further enhanced by distributed training.

---

[1]Corresponding author.

Contents

1 Introduction

Particles undergo complex stochastic interactions upon contact with materials. The modeling of these interactions is further complicated by the large number of secondary particles involved. The Monte Carlo simulation depends on repeated random sampling to produce a snapshot of the particle interactions with a detector. Simulation is crucial in most High Energy Physics (HEP) experiments and is extremely resource-intensive. More than 50% of the current computing resources of the HEP community are utilized in simulation alone [1]. In the future that need will increase further due to the higher luminosity and granularity of future experiments, and it will not be possible to create a corresponding increase in computing resources. The main motivation for the fast simulation is to incorporate other faster

alternatives to decrease the cost of future experiments. Current fast simulation approaches are mostly based on parametrization [2–4] or lookup table [5] approaches, providing between 10 and 100 times speedup while achieving different levels of accuracy.

Generative Adversarial Network (GAN) [6] is a training paradigm for deep generational neural networks. Other approaches for generative networks include Variational autoencoders [7] and autoregressive models [8], etc. All of these approaches have their own strengths and weaknesses. The autoencoder-based methods often produce blurry images while the pixel-based methods are not only slow to evaluate but also suffer from limited capacity. The GAN approach has been able to demonstrate highly realistic and sharp images as compared to other approaches [6]. There have been many recent variants of the GAN methodology, such as WGAN [9], StackGAN [10], and Progressive GAN [11] further enhancing the quality and resolution of generated images. The generative problem often involves intractable probability densities and thus methods involving likelihood estimates are not practical for most purposes. GAN can learn a distribution implicitly since it does not rely on the explicit computation of probability densities. Therefore the GAN approach is suitable for the generation of a wider range of data such as musical notes [12], natural language [13], medical data [14, 15], natural scenes [6], faces [11] and image denoising[16]. Since simulation is essentially a generative problem thus a successful image generation model can be exploited in this domain. A potential advantage of deep generative models is that the generated distribution does not have to be explicitly defined, and thus even the real data from a detector can be simulated directly.

We leverage the GAN methodology to generate HEP calorimeter output. Calorimeters are special HEP detectors that record particles through the measurement of the energies deposited by them. These detectors can be regarded as huge cameras taking pictures of particle interactions. The Monte Carlo simulation for the detector output is extremely precise but highly expensive both in regards to the simulation time and resources. For most HEP experiments the calorimeter is a simulation bottleneck, consuming, for example, more than 80% of the simulation time for the ATLAS experiment [2]. We generate the calorimeter cells as monochromatic pixelated images with the cell energy depositions as our pixel intensities.

The 3DGAN [17–19] is the first effort where the detector output is generated retaining correlations in all three spatial dimensions. We employ three-dimensional convolutional layers in our network, while previously detector output was generated either as a two-dimensional image or a concatenated set of two-dimensional images. We demonstrate our approach for a high granularity detector with higher spatial resolution and thus consequently much larger image dimensions than previous such efforts. We pre-process the cell depositions by taking a power less than one, thus decreasing the dynamic range of corresponding pixel intensities and improving the convergence. We employ a multi-step training process to generate images, from a complex multivariate distribution, for a large range of input conditions. We also perform extensive validations from diverse viewpoints including vision and deep learning, as well as, physics-based evaluation. The network scores highly on all the platforms both for the pertinence to the training data and for maintaining sufficient diversity. The details of the network development and the validation from different

perspectives have been presented previously [18]. The current work is geared more towards the physics community and thus is limited to only physics-based validation. Previously we simulated electrons coming with energies from a wide spectrum by employing our multi-step training. Now we successfully extend the same approach to simulate additional particle types such as photon and neutral pion where most of the energy is lost in electromagnetic interactions. We perform an additional investigation to prove that the GAN could accurately reproduce the signature features of a particular particle type. We also undertake some preliminary exploration of the charged pion simulation and generation of rare events. We finally present a successful practical example of using the GAN-generated data in a typical reconstruction tool, demonstrating that the GAN-generated images could provide similar performance as Monte Carlo images.

The current paper is organized as follows. Section 2 presents the overview of past efforts for fast simulation of HEP calorimeters exploiting deep neural networks. The next section (Section 3) describes the Monte Carlo training dataset. The basic structure of the calorimeter and the important features of our data are discussed. Section 4 describes how the GAN approach is adapted to the problem of HEP detector simulation. The approach has been exploited for the generation of particles with predominantly electromagnetic showers. The results for comparison to Monte Carlo simulation are presented in Section 5. Some preliminary work is also carried out for the charged pion simulation as presented in Section 6. Another study exploring the simulation of rare modes is presented in Section 7. Section 8 presents a practical application for the use of GAN-generated data. Finally, Section 9 summarises the main contributions and presents some future suggestions.

2 Previous Work

Fast simulation is already incorporated in existing experiments through approaches like parametrization [2–4, 20] and Lookup tables [5], etc. Usually a part of the simulation is replaced by fastsim where some tradeoff between speed and accuracy can be feasible. Following the same concept deep learning has also been explored to generate simulation data. Fast simulation using neural networks can be regarded as a special type of parametrization, with the weights of the neural network as parameters, optimized through a training process.

The GAN technique is an unsupervised training methodology. The power of GAN lies in the fact that the target distribution does not have to be tractable and instead the training relies on a Minimax game between a discriminator (D) network and a generator (G) network. The discriminator is trained to differentiate between the target and the generated distributions, while the generator is trained to confuse the discriminator. Both the networks compete with each other till the generator manages to completely confuse the discriminator, given enough capacity for both models. At this point, the GAN is said to have converged.

Calorimeter data have been simulated through deep generative networks in a number of recent approaches as presented in Table 1. LAGAN [21] was one of the first fast simulation approaches based on deep learning. A simplified calorimeter was simulated as 2D jet images for high energy W bosons (signal) and generic quark/gluon jets (background). CALOGAN [22] employed the LAGAN architecture to generate sets of three two-

Table 1. Calorimeter simulation with deep learning

No.	Model	Algorithm	Architecture	Condition	Output
1	LAGAN [21] (2017)	GAN	2D Locally connected	Particle Type as discrete labels	25×25
2	CALOGAN [22] (2017)	GAN	2D Locally connected	$E_P \sim U(1,100)$ GeV	layer1: 3×96 layer2: 12×12 layer3: 12×6
3	3DGAN initial prototype [17] (2018)	ACGAN	Conv3D	$E_P \sim U(2,500)$ GeV	$25 \times 25 \times 25$
4	ATLAS [23] (2018)	WGAN and VAE	Dense	$E_P \sim U(1,260)$ GeV	vector of 266 cells.
5	LHCb [24] (2019)	WGAN	Conv2D	Five variables related to position and momentum	30×30
6	HGCAL [25] (2019)	WGAN	Conv2D and Locally connected	E_P and initial impact position (x,y)	Concatenation of 7 (12×15) layers
7	3DGAN [19] (2019)	ACGAN	Conv3D	$E_P \sim U(2,500)$ GeV and $\theta \sim U(60°,120°)$	$51 \times 51 \times 25$
8	DijetGAN [26] (2020)	WGAN	Conv2D		vector of 7 jet variables.
9	ILD [27] (2021)	GAN, WGAN and BIB-AE	Conv3D	$E_P \sim U(10,100)$ GeV)	$30 \times 30 \times 30$.

dimensional images that were then concatenated to obtain the output for a three-layered simplified calorimeter conditioned on the primary particle energy (E_P) ranging from $1-100$ GeV. Since then, there have been other demonstrations employing deep learning for HEP calorimeter simulation. Deep learning has been used for fast simulation of the ATLAS calorimeter [23]. Showers with energies $1-260$ GeV and pseudorapidity $|\eta|$ in the range of $0.2-0.25$ were generated as flattened arrays of pixels, by a dense network employing both VAE and GAN methodologies. The images were also conditioned on the primary particle energy and constrained on the total energy deposition. The GAN-generated showers were reported to have better performance as compared to the VAE generated showers. WGAN

has also been used to simulate the LHC detector output collapsed to a two-dimensional array of cells [24]. A simplified version of HGCAL was simulated as seven 2D images concatenated together, conditioned on the primary particle energy and impact position [25]. The DijetGAN [26] employed GAN for the simulation of dijet events: a background process for important physics studies at LHC. A more recent approach [27] experimented with several GAN architectures for the simulation of high granularity calorimeter for the ILD [28] detector as $30 \times 30 \times 30$ three-dimensional images. The simulation was limited to the orthogonally incident photons coming with $10 - 100$ GeV primary energy.

The 3DGAN initial prototype [17] exploited 3D convolutional networks to simulate the response of a high granularity calorimeter as $25 \times 25 \times 25$ image. The GAN setup was used to train the network for a simplified scenario involving only orthogonally incident electrons. The approach was then extended to condition $51 \times 51 \times 25$ images on both the particle energy and incident angle [18, 19]. The more complex distribution could be generated through multi-step training, architecture, and loss function modifications (details in [29]). We now simulate the detector output for all the particle types available in the dataset and further validate the results. The 3DGAN greatly surpasses existing efforts in the granularity and dimensions of the generated images, conditioned on both the incident particle angle and energy from a wide range, and validated in great detail from diverse viewpoints. Finally, a culmination of the effort is to test the GAN-generated data for a practical use case.

3 Calorimeter Dataset

We present a solution for the needs of future experiments with higher demands for computing resources due to increased luminosity and granularity. We, therefore, select the proposed Linear Collider Detector (LCD), designed in the context of the future Compact Linear Collider (CLIC) [30] accelerator for our study. The dataset employs the GEANT4 toolkit [31] for the generation of the simulation data for several particle types (i.e., electrons e, photons γ, neutral pions π^0 and charged pions π) and is publicly available on Zenodo at https://zenodo.org/communities/mpp-hep.

3.1 Detector Geometry

Figure 1 shows the proposed detector design, highlighting the main detector concepts. The basic design consideration for improving the jet energy resolution is to resolve the energy depositions of the individual particles in a jet, through a high cell granularity and precise time information. The electron and positron will collide in the central region. The trackers are shown in blue. The surrounding grey region will comprise the calorimeters. The calorimeter will be highly segmented with an electromagnetic (ECAL) and a hadronic (HCAL) calorimeter.

The data used for the current work is that of the ECAL central barrel region. This region has a cylindrical shape with an inner radius of 1.5 m with 25 concentric layers. The proposed granularity for the ECAL cells is 5.1×5.1 mm^2. The cells are arranged in the form of 25 cylindrical layers with silicon sensor planes (active), alternating with tungsten absorber planes (passive). The simulation is carried out considering the entire detector
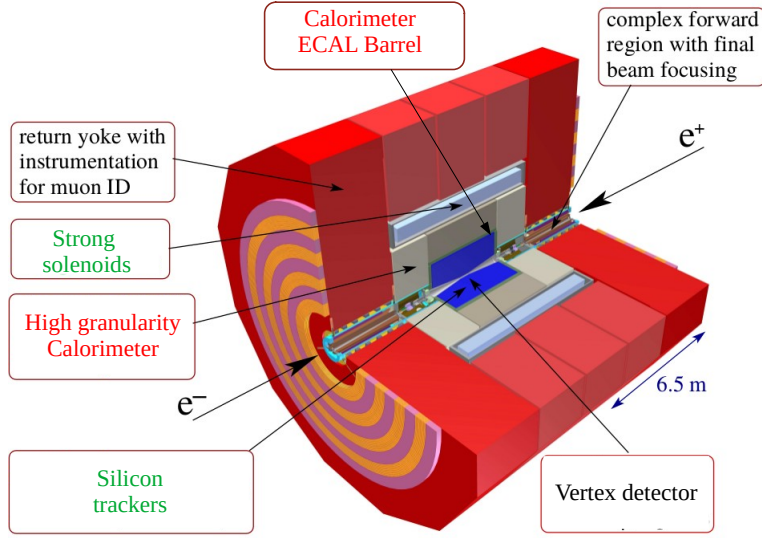
Figure 1. Schematic diagram for the CLIC calorimeter [32]
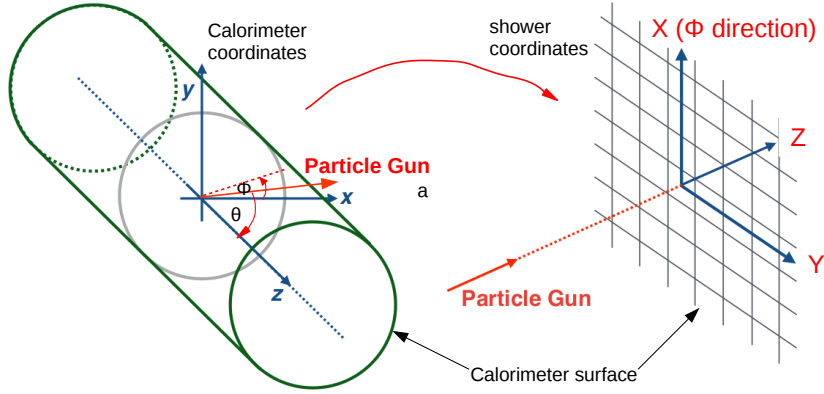

Figure 2. The calorimeter barycenter computation was done in global coordinates (left) and the cells were saved based on the local coordinates (right).

geometry, including the material in front of the calorimeter, and the effect of the solenoid magnetic field.

3.2   Data Features

The energy deposits in the calorimeter cells result from the interaction of an incoming primary particle with the calorimeter material. These deposits form a characteristic shape that can be termed as an "event" or "shower". A slice around the barycenter of each shower is saved as a 3D array of energy depositions. The slicing is carried out by taking a projection of all the deposited energy on the ECAL inner surface. The barycenter of this 2D image and the point of origin of the incoming particle are then used to compute the polar angles (θ and ϕ) corresponding to each shower. Due to the different ϕ granularity for each depth

layer in the ECAL, a multiplicative transformation is also applied to scale every layer to look like the innermost ECAL layer. Finally, the data is saved in the HDF5 format. Each entry in the dataset comprises the 3D array with cell energy deposits, the incoming particle energy E_P, and the incidence angles θ and ϕ. The energies for the incoming particles are uniformly distributed from 2 to 500 GeV and an incident angle (θ) uniformly distributed from $60°$ (1.047 radians) to $120°$ (2.094 radians).

Figure 2 shows the particle gun position with respect to the calorimeter surface. In the global coordinate system, the Z axis lies along the axis of the calorimeter cylinder. While in the local coordinates of the shower it is perpendicular to the calorimeter surface. Similarly, other axes are also transformed to the local coordinates of each sample. The Z axis of our 3D images lies along with the detector depth and X, Y are the transverse axes. The calorimeter is isotropic in the ϕ direction, therefore only the θ direction is incorporated for the current work. The θ value is recomputed as a weighted mean of the angles computed using the barycenter of the event and the barycenters of the XY planes for each position along Z (weighted by the position along Z).

4 3DGAN

The HEP simulation depends on a set of variables that impact the underlying physics processes described by the simulation. Therefore, 3DGAN uses the E_P and θ of a particle striking the calorimeter surface as inputs, to generate the appropriate detector response. In order to provide feedback on the correspondence between generated showers and input conditions, we exploit the concept of auxiliary tasks [33] together with domain-related constraints. The current works mainly describe the final optimized version of the 3DGAN model while additional details about the development process can be obtained from [29]. The 3DGAN is implemented using Keras 2.2.4 [34] deep learning python library with Tensorflow 1.14.0 [35] as a backend. The code is available at `https://github.com/svalleco/3Dgan`.

4.1 Pre-processing

One of the main challenges for generating scientific data through techniques developed for computer vision lies in the inherent difference between the dynamic ranges of the pixel intensities. The pixel intensities in a typical RGB image have a range from 0 to 255 while the energy deposited in detector cells covers more than 10 orders of magnitude. We have implemented a pre-processing procedure aimed at reducing this dynamic range. Initial tests conducted taking the logarithms of the pixel intensities, resulted in the generation of highly distorted images. Taking a less drastic approach we calculate the power function of pixels intensities using an exponent smaller than one. A smaller exponent results in faster convergence but greater distortion in generated images, while a larger exponent slows down convergence yet retaining image quality. Figure 3 shows how the value of the exponent (p) affected the distribution of the generated pixel intensities for the individual cells. The value of p is adjusted to an optimum value of 0.85, where a faster convergence is achieved while retaining an acceptable level of accuracy at both ends of the spectrum. The generated images are then post-processed by simply taking the inverse of the power function. The
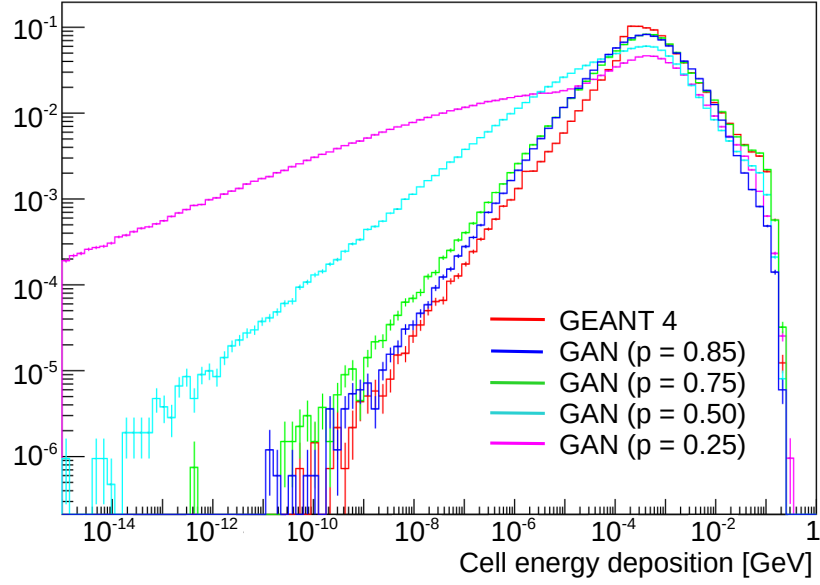
Figure 3. The cell energy distribution for GEANT4 MC events (red) and GAN generated events after a pre-processing by taking the power of pixel intensities: power =0.85 (blue); power =0.75 (green); power =0.5 (cyan); power =0.25 (magenta).

data is also subjected to a threshold of 0.2 GeV for the total energy deposited in the event. This rejection is aimed at removing some spurious events with little or no energy deposition.
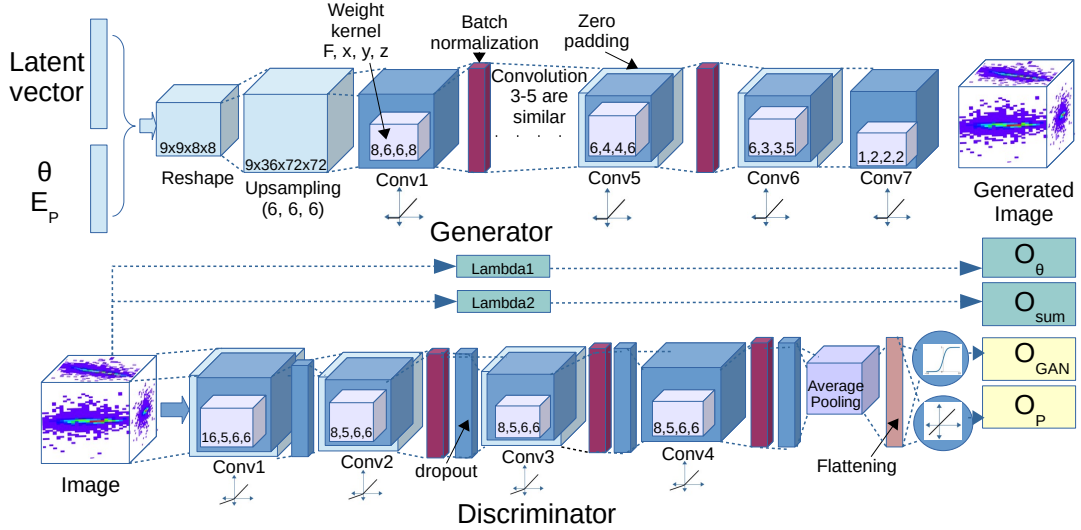
4.2   Architecture

Figure 4. The 3DGAN architecture, see the text for details.

The 3DGAN architecture is presented in Figure 4. The generator network implements

stochasticity through a latent vector of 254 random numbers drawn from a Gaussian distribution. The generator input includes E_P and θ concatenated to the latent vector. The generator network then maps the input to a layer of linear neurons followed by seven 3D convolutional layers. The discriminator input is an image while the network has only four 3D convolutional layers. Batch normalization [36] is performed after all except the first convolutional layer in the discriminator and the last two layers in the generator. The leakyRelu [37] activation function is used for the discriminator hidden layers while the Relu [38] activation function is used for the generator layers to induce sparsity. The discriminator uses dropout [39] of 20% for regularization and a single average pooling layer after the last convolutional layer since additional pooling layers result in substantial loss of performance.

The discriminator network has two trainable outputs: a sigmoid neuron predicts the O_{GAN} and a linear neuron O_P predicts E_P. The other two additional outputs are simple analytical measurements: O_{sum} is the total deposited energy and O_θ is the measured incident angle (geometrical angle of the shower energy depositions). These non-trainable outputs represent physics-based constraints.

4.3   Loss Function

The 3DGAN loss function is the weighted sum of individual losses pertaining to the discriminator outputs and constraints. The domain-related constraints are essential to achieve a high level of agreement over the very large dynamic range of the image pixel intensity distribution. Equation 4.3 presents the discriminator loss related to the output O_{GAN} as L_G, the loss related to the output O_{sum} as L_E, the output O_θ as L_A and the predicted E_P as L_P balanced by corresponding weights W.

$$L_{3DGAN} = W_G L_G + W_P L_P + W_A L_A + W_E L_E \qquad (4.1)$$

The L_P and L_θ both provide feedback on how well the generated images correspond to the input conditions. The loss L_E ensures energy conservation. L_G is evaluated as binary cross-entropy. L_P and L_E are implemented on mean percentage errors, while L_A as mean absolute error. The generator loss is implemented as the inverse of L_G together with the auxiliary losses and constraints. The weights (presented in Appendix A) are considered as hyperparameters and chosen to balance the loss ranges and their relative importance (in this case the loss L_G is given higher priority as compared to the auxiliary losses).

4.4   Training

The 3DGAN training is inspired by the concept of transfer learning. The GAN could not converge for the highly complex multivariate distribution directly thus a two-step training is applied. In order to successfully train the network, we reduce the complexity by training the GAN first for electron events having $E_P \sim U(100, 200)$ GeV. After the GAN converges, the same trained model is further trained with the data from the whole E_P range of $E_P \sim U(2, 500)$ GeV. The first training step exploits $137, 342$ electron events. The GAN

is then trained for the larger E_P range, utilizing a much larger size of training data (400 k events) from each particle type (electrons, photons, and neutral pions). The train and test losses are evaluated on the data divided in a ratio of nine to one. The first training step is run for 130 epochs (2 hours per epoch on GTX 1080) while the second step is run for 30 epochs (4 hours per epoch on GTX 1080). Finally, the best network is selected according to the minimum relative error for the SF on additional validation data (20k events are filtered around specific E_P bins). This last step is aimed to further improve the accuracy for the SF.

Algorithm 1: Training the 3DGAN model

---

initialize
for *number of epochs* **do**
    #Training
    for *all batches in the training data* **do**
        get real E_P, θ, E_{sum} and image batches from data
        latent batch $\sim \mathcal{N}(0,1)$
        generator input $= E_P \mathbin{+\!\!+} \theta \mathbin{+\!\!+}$ latent
        generate fake events for the same E_P and θ
        train discriminator on real batch
        train discriminator on fake batch
        for *2 times* **do**
            latent batch $\sim \mathcal{N}(0,1)$
            generator input $= E_P \mathbin{+\!\!+} \theta \mathbin{+\!\!+}$ latent
            use real E_P and θ for fake events
            train the generator (minimizing the discriminator loss on the generated
              images)
        end
        # Testing
        Evaluate discriminator on real and fake data
        Evaluate generator using the inverse of discriminator loss on generated data
    end
end

The training process for each epoch is presented by the Algorithm 1. For each training iteration the discriminator is trained twice: once on a batch of real data, and next on a batch of generated data. For a balanced approach, the generator is also trained twice while freezing the weights of the discriminator. The RMSProp [40] optimizer is utilized to train the network through Stochastic Gradient Descent. Figure 5 shows the L_G losses associated to the discriminator (blue) and the generator (orange). It can be seen that the loss for the discriminator increases, while the loss for the generator increases till both losses are converged at around 0.6 (log(4)). At this point, the O_{GAN} output for both the data (red) and the GAN images (blue), shown in Figure 5 right panel has similar distributions centered around 0.5. The discriminator is indeed confused and the GAN converges.
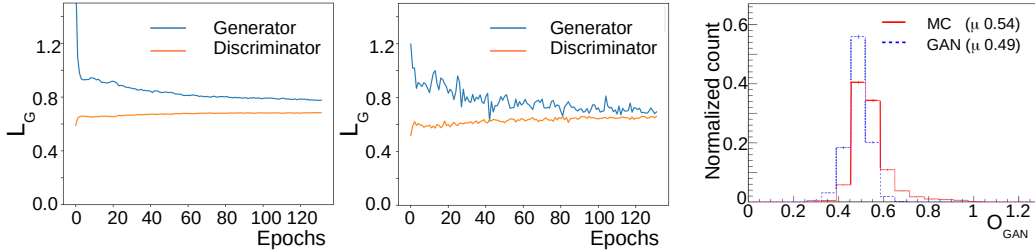
Figure 5. The L_G losses for the generator (blue) and the discriminator (orange) converge for the first training step with restricted E_P range. (Left) training losses. (Middle) test losses. (Right) the discriminator output O_{GAN} for MC (red) and GAN (blue) images is similar and the discriminator is confused.

Table 2. Inference Timing for 3DGAN

Method	Platform	Time/shower (msec)	Speedup
Monte Carlo		17000	1.0
3DGAN CPU	2S Intel Xeon Platinum 8180	16	2500
3DGAN GPU	GTX 1080	4	4250

4.5 Generation Time

The 3DGAN greatly reduces the simulation time. Table 2 compares the time taken to generate a shower using Monte Carlo and GAN. The inference time is around 4 ms/particle on GeForce GTX 1080. A similar shower can be simulated using the GEANT4 in about 17 second per particle on an Intel Xeon 8180 (currently it is not possible to run a full Geant4-based simulation on GPUs). The speedup of many orders of magnitude is achieved.

5 Results and Discussion

The performance assessment for GAN models is a subject of much debate and diverse viewpoints [41]. The GAN evaluation is nontrivial due to the intractable probability densities and thus is mainly sample-based and application-specific. We have validated the realism and diversity of our generated data from several independent viewpoints, such as the output of a third-party neural network and image quality assessment as presented previously [29] but the current work focuses mainly on the physics-based comparison to a Monte Carlo simulation.

The particle showers have specific characteristics due to the underlying physics processes, depending on the detector material and the type, energy, and direction of the particle initiating the shower. We validate these characteristics as a function of our inputs by dividing the data in 5 GeV E_P and 0.1 radian (5.73°) θ bins. To ensure an unbiased comparison, GAN events are generated with the same E_P and θ values as the GEANT4 events. The bin-wise comparison of each physics-based feature, results in hundreds of histograms, for each particle type. We present here a selected subset of the detailed and exhaustive
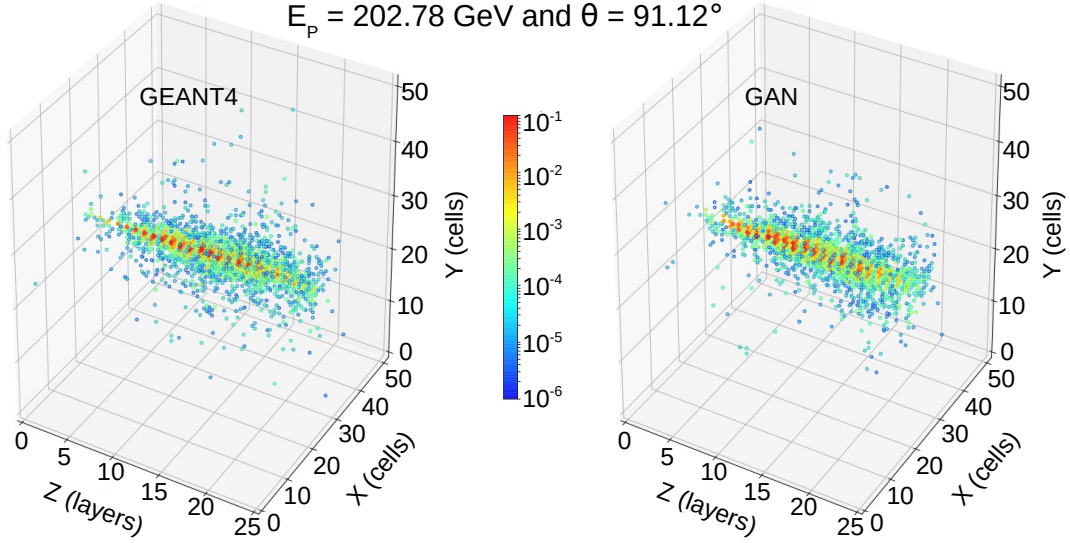
Figure 6. GEANT4 vs. GAN electrons showers with $E_p = 202.78$ GeV and $\theta = 91.12°$.

validation that we consider to be the most essential and representative of performance. The results presented in this section are also alternated among different bins and particle types, in order to convey the overall level of accuracy.

5.1 Visual Inspection

An initial qualitative assessment can be performed by comparing the events visually. Figure 6 shows an example of a 3D electron event. The event on the left has been generated by GEANT4, while the event on the right has been generated by the GAN for the same input values. It can be observed that both events have very similar visual characteristics while retaining uniqueness for individual cell deposits. The graphical projections on different planes further illustrates the shower correlation to the incident angle (θ) and energy (E_P). Figure 7 compares the projections of the GEANT4 showers to the corresponding GAN showers. The top panel presents neutral pion events, while the bottom panel display photons events, with the E_P and θ from both ends of the spectrum. The GAN images appear similar to the respective GEANT4 images with the deposited cell energies correlated to the input conditions while retaining stochasticity, for all particle types and input conditions.

5.2 Particle Shower Features

Shower shapes define the structure of the deposited energy distribution, as a shower develops through the detector material. The profiles of the energy deposition along the detector axes are important observables related to the shower geometry and crucial for most particle identification techniques. We would like to point out that these geometrical features are not included in the 3DGAN loss function as presented in Equation 4.3 and are learned by the GAN implicitly. Figure 8 presents the shower shapes for the X, Y and Z axes as a function
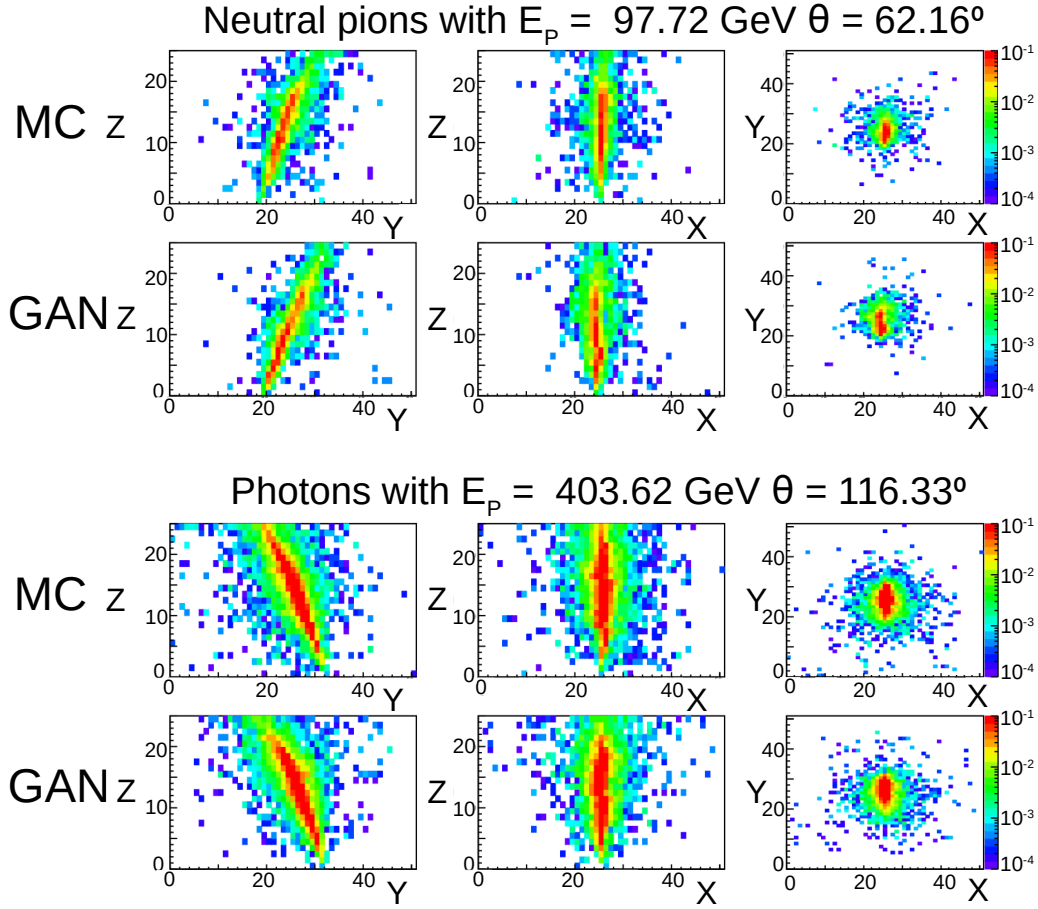
Figure 7. Projections of Monte Carlo vs. GAN events on the YZ, XZ and XY planes. Top) neutral pions with $E_P = 97.72$ GeV and $\theta = 62.16°$. Bottom) photons with $E_P = 403.62$ GeV and $\theta = 116.33°$.

of θ and E_P. In order to summarize the performance for all particle types, we present a different particle in each column: the shapes for electrons in the first column from the left, the photons in the middle column, and the neutral pions in the rightmost column. The top row presents the transverse shape distribution for the Y axis corresponding to the different θ bins since the Y axis profile is most relevant for θ. The plots are displayed in the log scale to enhance the sparse distributions along the tails. The second row presents the shapes along the Z axis (longitudinal direction) in linear scale for the different E_P bins. The network is able to reproduce a similar shape distribution as the GEANT4 showers, furthermore, the network can correctly relate it to the inputs. In the transverse profiles, some discrepancies are observed in the log scale. These discrepancies occur at the volume edges, where smaller energy depositions occur. This region is also highly sparse and outside the main body of the shower with expected energies well below 0.1 MeV, which is comparable to the pedestal values.

Moments are another aspect of the shower geometry. The GEANT4 showers are all
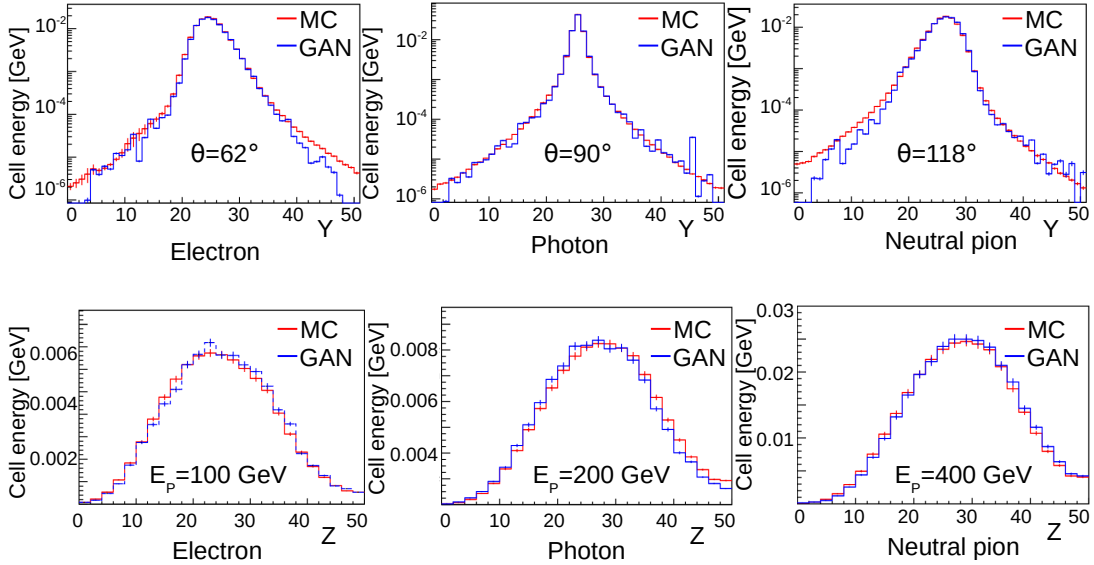
Figure 8. Shower Shapes for the GEANT4 vs. GAN events as a function of inputs. Top row) longitudinal shower shapes along the Y axis: left) electrons with θ in the 62° bin; mid) photons with θ in the 90° bin; right) neutral pions with θ in the 118° bin. Bottom row) transverse shower shapes: left) electrons with E_P in the 100 GeV bin; mid) photons with E_P in the 200 GeV bin; right) neutral pions with E_P with the 400 GeV bin.

centered on the barycenter of the energy deposition by design thus the first moment ($M1$) defining the shower center is easily replicated by GAN. Therefore, we present here the performance related to the second moment ($M2$) having a more complex distribution depending on both E_P and θ. Figure 9 left panel presents the distribution of the second moment or the width of the shower for electrons. Here it can be appreciated that the GAN has learned the non-Gaussian width distribution. The mid plot shows the difference between internal correlation present between the shower inputs (E_P and θ) and the shower features (shapes, moments, total deposition, hits, and ratios of energy deposition in different parts of the shower) for GEANT4 and GAN photons. The GAN showers are able to reproduce the internal correlations present between the different shower observables. The right panel displays the close agreement between the θ measured from GEANT4 neutral pion events and that measured from the GAN events generated for the same θ values.

The energies deposited in detector cells are the pixel intensities of our images. The images are mostly empty, centered around a shower. The energies are deposited only in around 20% of the cells. Figure 10 left panel shows sparsity (S) as the fraction of cells with some deposition against the threshold used for cutoff. The GAN images have similar sparsity distributions as the GEANT4 events, without specifically constraining the image. The distributions for cell energy depositions (photon) shown in the mid panel have a similar shape for GEANT4 and GAN events. We had reported that a sharp, vertical drop around 0.2 MeV was present in the GEANT4 cell energy distribution that the GAN could not learn, yet tried to smooth out in the best manner [18]. Since then a recent work [27] also recognizes this feature and tries to replicate the effect by additional post-processing on the
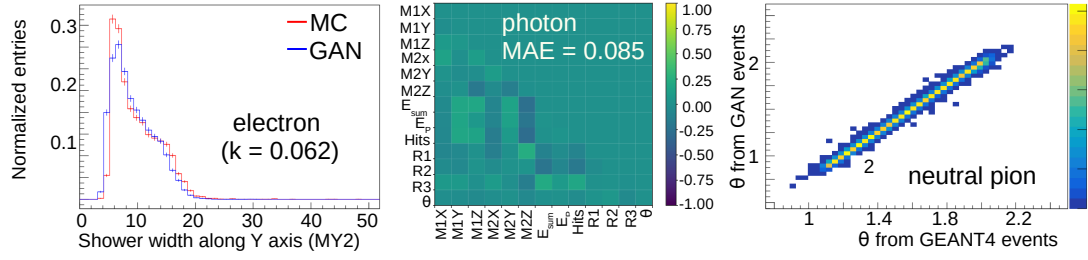
Figure 9. The GAN vs. GEANT4 shower features. Right) the shower width (M2) along the Y axis for electrons. Mid) difference between internal correlations present between physics features and the inputs for photons. Left) the correlation between the measured angle from GEANT4 events and the GAN events generated for the same θ values for the neutral pions.
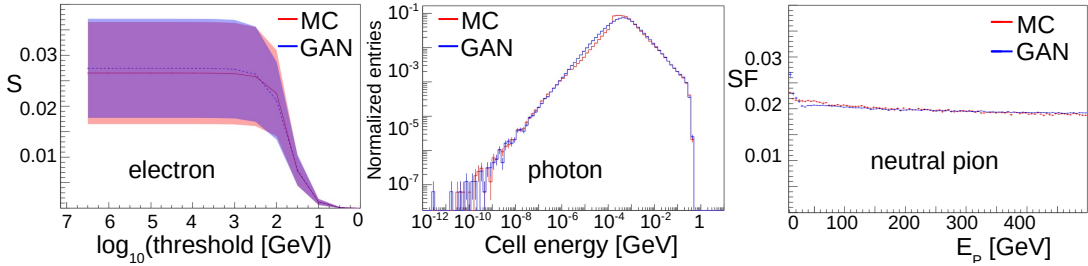

Figure 10. Shower features related to the pixel intensities (cell depositions) for the GEANT4 (red) vs. GAN (blue) events. Left) sparsity level as a function of cutoff threshold for electrons. Mid) distribution of cell energy deposits for photons. Right) sampling fraction for neutral pion showers.
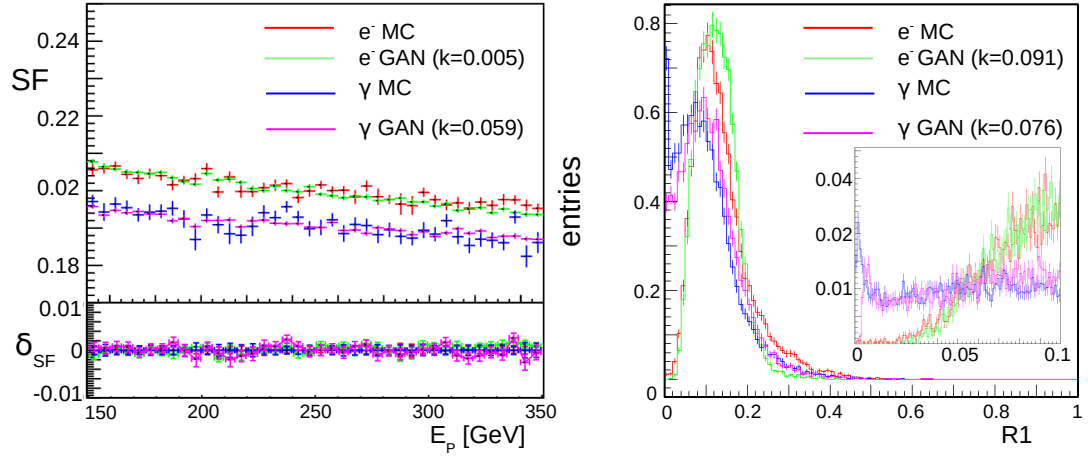

Figure 11. The GAN could successfully learn the photon features after further training a network generating electrons with reduced E_P range. Left) the SF for photons is lower than electrons both in GEANT4 events and GAN events. The bottom panel shows the relative error (δ_{SF}) for the sampling fraction. Right) the distribution of the energy fraction deposited in the first part of the shower ($R1$) for the electron and photon showers. The distribution for $R1 < 0.1$ (zoomed in the inset) has very different distributions for electrons and photons. The GAN also demonstrates a similar behavior.
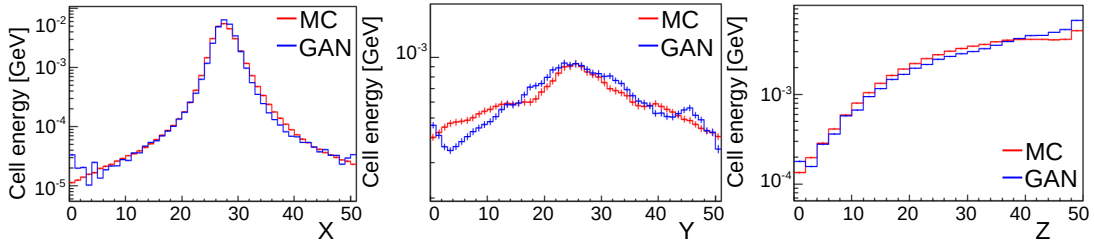
Figure 12. Shower Shapes for GEANT4 vs. GAN charged pion events along X (left), Y (mid) and Z (right) axes with random E_P and θ.

generated images. They note that their network improves the performance on the simulation of the correct pixel distribution at the cost of reduced performance for other features. We also report a reduction in performance when constraining the pixel intensity distribution through our loss function. We believe in the future the concept of ensembling [42] can be used to employ two networks to generate the pixels above and below this region. The sampling fraction (SF) is an important characteristic of the detector response. Figure 10 left panel presents SF for neutral pions. The SF is presented as a function of E_P. There is a close agreement for most of the input range with some discrepancies at low energies, where events are highly sparse with low cell energy deposits.

The characteristic features for the electromagnetic showers are faithfully reproduced in the GAN-generated showers for different particle types. We test if the generated showers for different particle types are mutually distinguishable through their corresponding features. This is crucial as the final networks for all particle types, use the same initial weights trained to generate electron-induced showers for a reduced energy range. The photon-initiated showers have some minor differences from electron-initiated showers. The photons penetrate more distance into the detector material before starting to interact [43, 44]. This effect can be evaluated by studying the SF and the fraction of energy deposited in the first (8 cells along the Z axis) part of the shower. Figure 11 left panel shows the profiles of SF as a function of E_P for the central region of the spectrum. The SF for photons is lower than electrons for a similar value of E_P. The right panel compares the distribution of energy deposited in the first part of the shower for electrons and photons. The photons present more entries for the region where the fraction of energy deposited in the first part of the shower $R1$ is less than 10%. The GAN-generated photons clearly demonstrate these identifiable features.

6   Simulating Charged Pions

The charged pions deposit a much smaller part of their energies in the ECAL while most of the energy is deposited in the HCAL. The current project is only limited to the ECAL data due to the limitation of the computing resources, thus the work for charged pion is only a preliminary study. A more accurate approach will also need to incorporate data from HCAL. We will present the results of our study to lay the foundation for any future work.
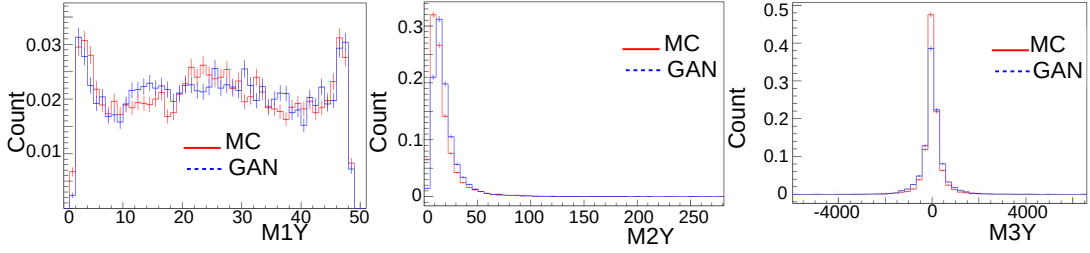
Figure 13. Shower moments along Y axis for GEANT4 vs. GAN charged pions. Left) $M1Y$ (shower center). Mid) $M2Y$ (shower width). Right) $M3Y$ (shower skewness).
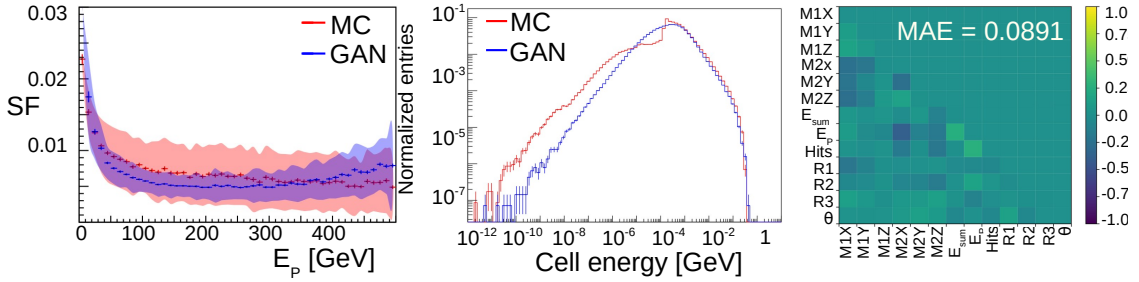

Figure 14. Shower features for GAN vs. GEANT4 charged pion events. Left) SF as a function of E_P with shaded area representing standard deviation. Mid) pixel intensities with log yscale. Right) difference between internal correlations among shower features.

The transfer learning approach could not be extended to charged pions as the showers have a very different distributions. The GAN is trained for the full E_P range of the charged pions, directly from random weights, for about 200 epochs using $(300k)$ events. There is a great diversity in the charged pion events, and most of the events have low energy deposition in the ECAL. Since visual inspection will not be helpful to understand the performance, thus we present only the distributions of physics-based features. Figure 12 presents the overall shower shapes in log scale. The shape distributions along the X and Z axis show slightly better performance as compared to the Y axis, probably due to the higher variance present in this dimension. Figure 13 compares the distribution of the first three moments defining the shower center, width, and skewness along the Y axis, display similar distributions.

Figure 14 left panel displays the sampling fraction for GEANT4 and GAN charged pion events. It can be seen that there is a difference between the means of the two distributions, while the shaded area representing the standard deviation shows some overlap. The mid panel presents the distribution of the pixel intensities that are energies deposited in the calorimeter cells. There is a deterioration in performance as compared to other particles, particularly the effect of the presence of a cut around 0.2 MeV (see Section 5) is more pronounced. The left panel presents the difference in internal correlations between physics-based features like shapes, moments, hits, E_P and θ for the GEANT4 and GAN events. There is a less than 10% error for the correlations, even given the highly diverse and incomplete showers. In order to convey an idea of the diversity present in these showers, we investigate the barycenters of the shower energy depositions along the longitudinal axis (Z)
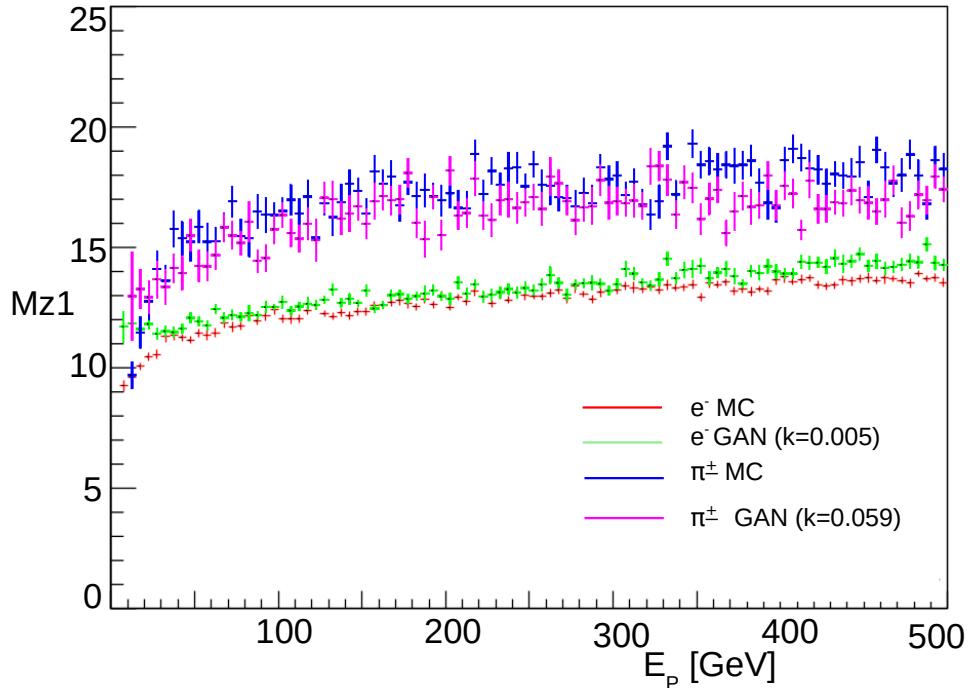
Figure 15. The distribution for the first moment along the Z axis (Mz1) for electrons and charged pions. The barycenter of deposition along the Z axis shows very different distributions for both particles with the charged pion showers starting much later.

for electrons and charged pions, as a function of E_P. Figure 15 compares the first moment for the GEANT4 and GAN events. The charged pion showers show greater diversity and depth as compared to electrons, also apparent in the GAN events.

7   Rare modes

The discriminator assigns a higher probability of being realistic to images exhibiting features that the GAN cannot reproduce correctly . We visually investigate such events having O_{GAN} value greater than 0.6, with the help of graphical projections. It is observed that most of these events manifest rare modes in data like pre-showering, late showering, and incorrectly centered events. As other modes are found to be even rarer, thus only the early showering events are further investigated. Figure 16 top row presents an example of a GEANT4 pre-showering event. The particles that start depositing their energies before entering the calorimeter volume have multiple particles striking the detector surface, thus resulting in multiple branches. Only a few percent of the Monte Carlo samples present such behavior and the percentage decreases with increasing E_P.

Figure 16 bottom row presents an example of a pre-showering event generated by GAN. It must be mentioned here that in the first training step more such events are generated. The performance for these rare modes deteriorates with further training for the full E_P range due to a decrease in the percentage of such events for higher energies. In order to further improve the performance, methods like the ensembling [42] can be explored.
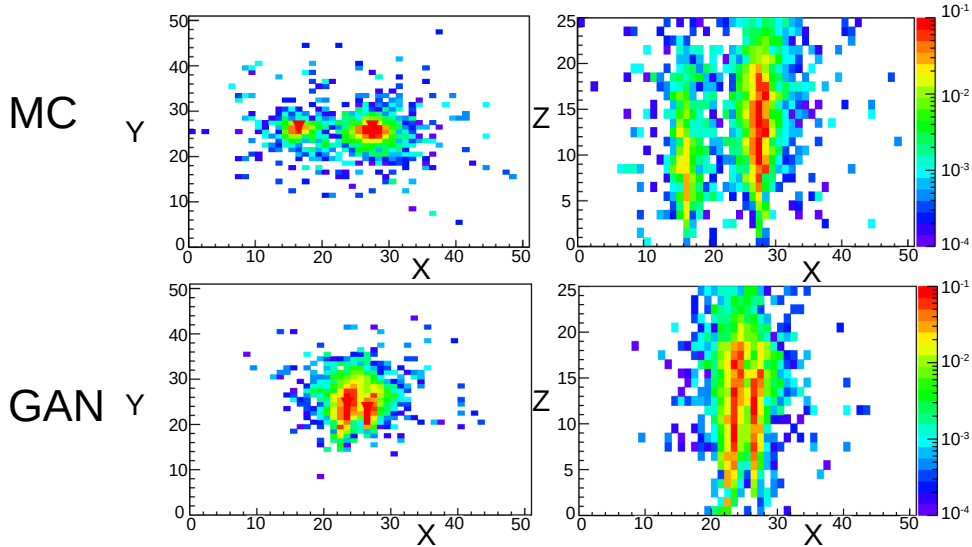
Figure 16. Graphical projections for the XY and XZ planes for MC events with high discriminator probability of being real images; top) GEANT4 event; bottom) GAN event.

8 Training with GAN data

We present a practical use case for the GAN-generated events. Triforce [19] is a deep learning model developed by a third-party study for the identification of particle type and primary energy for particle showers from the calorimeter dataset used for 3DGAN. We had previously employed the pre-trained Triforce DNN model for classification and regression of the GAN events [18] and proved that the type and the primary particle energy for the GAN-generated events was correctly predicted.

Triforce can also be considered as an example of a typical reconstruction tool used in HEP simulation. We test the performance of our 3DGAN generated images for training this tool. The Triforce requires two types of particles (electrons and charged pions) for training. We train the Triforce GoogleNet model from scratch on GEANT4 electron events and then on GAN electron events. The charged pion events in both trainings are those generated by the GEANT4. Figure 17 presents a comparison for the primary particle energy regression for the network trained on GEANT4 against the network trained on GAN electrons showing a similar performance. The particle type classification accuracy presented in Table 3 also manifest similar values. Thus we prove that the GAN simulation can be used to replace the GEANT4 simulation without any loss of accuracy.

In the context of GAN evaluation, the GAN-train and GAN-test are two very interesting concepts [45]. The accuracy of a classifier network trained on GAN generated events and tested on data events is termed as GAN-train. When GAN images are high quality and as diverse as the training set, the score on the validation set should be similar to training accuracy. A lower accuracy would indicate that GAN images are not covering the entire distribution of the training data. GAN-test is the accuracy of a network trained on true data and validated on GAN images. A lower accuracy would indicate that the GAN images
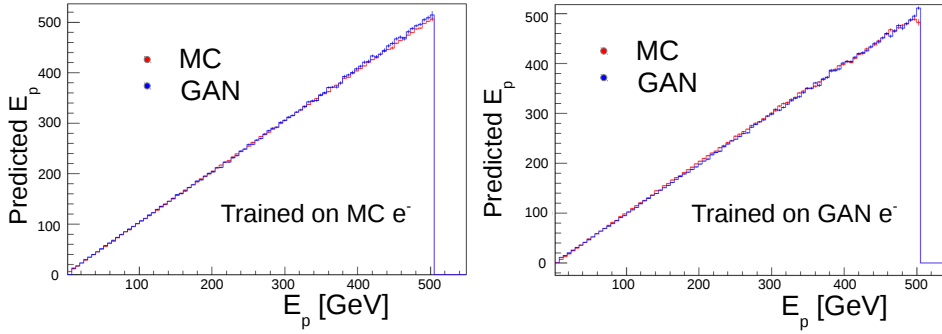
Figure 17. Performance of the network trained on GEANT4 electrons and charged pions (left) vs. that of the network trained GAN electrons and GEANT4 charged pions (right). The profile plot for true and predicted E_P for the GAN and Monte Carlo samples are very similar. There were 9834 events for each type.

Table 3. Triforce classification results

Trained on	Accuracy for MC	Accuracy for GAN
Monte Carlo electrons	0.9928 ± 0.014	0.9992 ± 0.014
GAN electrons	0.9884 ± 0.014	0.9998 ± 0.014

are not sufficiently realistic while a higher accuracy could be related to mode dropping. The 3DGAN shows similar performance as the GEANT4 training data and thus the results of this test can also be regarded as proof of high accuracy and diversity of the GAN generated events.

9   Conclusions and Future Suggestions

Simulation is crucial for most HEP experiments. Monte Carlo methodology can successfully simulate particle interactions at the cost of time and resources. Fast simulation is a set of faster alternatives that have been successfully used to replace detailed simulation where some loss in accuracy can be acceptable. Recent advances in deep learning have had a tremendous impact on the HEP community and are especially interesting for simulation due to the possibility of training directly from the detector data, not possible using other methodologies. The 3DGAN is an effort aimed to simulate the detector output, as images generated by a neural network. These images are conditioned on a number of variables, having large dynamic range of values. We exploit a multi-step training process, resulting in accurate simulation for electrons, photons, and neutral pions. The accuracy for individual data features varies but is within 10% of the GEANT4 simulation for all quantities. Preliminary work on charged pion simulation is also promising where the ECAL contains only partial showers, manifesting higher variance. The GAN is able to reproduce the essential features of a charged pion shower. Another exploratory study demonstrate the possibility of generating rare modes present in the data. We further provide an example of a practical use of the GAN-generated events. The GAN simulated events are able to train a third-

party particle classification and regression tool for the correct classification of the GEANT4 events. The response of the tool trained on GAN data is similar to that trained on GEANT4 data. Finally, we would like to state that the GAN-generated showers are simulated with a speedup of three orders of magnitude.

We would like to point out some insights from our endeavour that maybe helpful in the context of any future work. Certain domain-related features like the total deposited energy need to be hardcoded in the loss function while other features can be learned implicitly. These features include geometrical properties such as shapes and moments, level of sparsity, pixel intensity distribution, and correlation among the different features as well as the inputs. The model can learn complex distributions for individual features and the only part where the GAN struggles is reproducing the sharp drop in pixel intensities as discussed in Section 5. Apart from that, the sparse peripheral regions of the images are more difficult to be correctly generated, and there is some loss in performance for very low E_P particles. The current training allows learning of an average response for the total deposited energy since the training relies on direct comparison for small data batches. We believe that a better formulation of the loss might result in better agreement in future efforts. The preliminary work on charged pion and rare mode simulation shows great promise. The charged pion simulation can be improved by including the HCAL data. The generation of rare modes can benefit by exploring methods like the ensembling approach, where multiple networks can be trained simultaneously for different modes present in the data. The speedup can also be further increased by exploiting parallel hardware [46] that cannot yet be done for the sequential logic employed by the standard Monte Carlo tools. A distributed training [47] will be most essential for future generalization of the approach through hyperparameter scan.

Acknowledgments

A   Hyper-parameters for 3DGAN

Table 4 presents the values for the different hyperparameters selected for 3DGAN.

Table 4. Hyperparameters for 3DGAN fully optimized version

No.	param	value	description
1	epochs	130 for step 1 and 30 epochs for step 2	Number of iterations through entire training data
2	batch size	64	Number of samples in a mini-batch
3	latent size	254	Size of latent vector sampled from Gaussian with mean=0 and std =1
4	discriminator layers	4	Convolutional layers in the discriminator
5	generator layers	7	Convolutional layers in the generator
6	optimizer	RMSprop [40]	type of optimizer
7	lr	0.01	learning rate
8	pre-processing cell intensities by power	0.85	taking a power of cell intensities
9	pre-processing target E_p	scaling by 1/100	Dividing E_p by a factor of 100
10	W_G	3	Weight for loss associated to real/fake probability
11	W_P	0.1	Weight for loss associated to auxilliary energy regression task
12	W_E	0.1	Weight for loss associated to sum of intensities
13	W_A	25	Weight for loss associated to measured angle

References

[1] The HEP Software Foundation, J. Albrecht et al., *A Roadmap for HEP Software and Computing R&D for the 2020s*, *Computing and Software for Big Science* **3** (2019) 7.

[2] W. Lucas, *Fast simulation for atlas: Atlfast-ii and isf*, in *International Conference on Computing in High Energy and Nuclear Physics*, vol. 396, 2012.

[3] D. Orbaker, *Fast simulation of the cms detector*, in *International Conference on Computing in High Energy and Nuclear Physics*, vol. 219, 2010.

[4] NOMAD COLLABORATION collaboration, *Parameterization of e and γ initiated showers in the NOMAD lead-glass calorimeter*, *Nucl. Instrum. Methods Phys. Res., A* **425** (1998) 188. 28 p.

[5] E. Barberio et al., *Fast simulation of electromagnetic showers in the ATLAS calorimeter: Frozen showers*, *J. Phys. Conf. Ser.* **160** (2009) 012082.

[6] I.J. Goodfellow et al., *Generative adversarial nets*, in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, (Cambridge, MA, USA), pp. 2672–2680, MIT Press, 2014, http://dl.acm.org/citation.cfm?id=2969033.2969125.

[7] D.P. Kingma and M. Welling, *Auto-encoding variational bayes.*, in *ICLR*, Y. Bengio and Y. LeCun, eds., 2014, http://dblp.uni-trier.de/db/conf/iclr/iclr2014.html#KingmaW13.

[8] A. Van Den Oord, N. Kalchbrenner and K. Kavukcuoglu, *Pixel recurrent neural networks*, in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, p. 1747–1756, JMLR.org, 2016.

[9] M. Arjovsky, S. Chintala and L. Bottou, *Wasserstein gan*, 2017.

[10] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang et al., *Stackgan++: Realistic image synthesis with stacked generative adversarial networks*, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **41** (2019) 1947.

[11] T. Karras, T. Aila, S. Laine and J. Lehtinen, *Progressive growing of gans for improved quality, stability, and variation*, *CoRR* **abs/1710.10196** (2017) [`1710.10196`].

[12] S. Li, S. Jang and Y. Sung, *Automatic melody composition using enhanced gan*, *Mathematics* **7** (2019) .

[13] S. Subramanian, S. Rajeswar, F. Dutil, C. Pal and A. Courville, *Adversarial generation of natural language*, in *Proceedings of the 2nd Workshop on Representation Learning for NLP*, (Vancouver, Canada), pp. 241–251, Association for Computational Linguistics, Aug., 2017, DOI.

[14] X. Yi, E. Walia and P. Babyn, *Generative adversarial network in medical imaging: A review*, *Medical Image Analysis* **58** (2019) 101552.

[15] 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*, *Biomedical Signal Processing and Control* **55** (2020) 101632.

[16] A. Alsaiari, R. Rustagi, A. Alhakamy, M.M. Thomas and A.G. Forbes, *Image denoising using a generative adversarial network*, in *2019 IEEE 2nd International Conference on Information and Computer Technologies (ICICT)*, pp. 126–132, 2019.

[17] G. Khattak, S. Vallecorsa and F. Carminati, *Three dimensional energy parametrized generative adversarial networks for electromagnetic shower simulation*, in *2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 3913–3917, Oct, 2018, DOI.

[18] G. Khattak, S. Vallecorsa, F. Carminati and G.M. Khan, *Particle detector simulation using generative adversarial networks with domain related constraints*, in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pp. 28–33, Dec, 2019, DOI.

[19] D. Belayneh et al., *Calorimetry with deep learning: Particle simulation and reconstruction for collider physics*, *Eur. Phys. J. C* **80** (2019) 688. 31 p [`1912.06794`].

[20] ATLAS Collaboration, *The atlas calorimeter simulation fastcalosim*, in *IEEE Nuclear Science Symposuim Medical Imaging Conference*, pp. 1–5, 2010.

[21] L. de Oliveira, M. Paganini and B. Nachman, *Learning particle physics by example: Location-aware generative adversarial networks for physics synthesis*, arXiv preprint arXiv:1701.05927 (2017) .

[22] 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* **97** (2018) 014021.

[23] D. Salamani et al., *Deep Generative Models for Fast Shower Simulation in ATLAS*, in *Proceedings, 14th International Conference on e-Science: Amsterdam, Netherlands, October 29-November 1, 2018*, p. 348, 2018, DOI.

[24] V. Chekalina, E. Orlova, F. Ratnikov, D. Ulyanov, A. Ustyuzhanin and E. Zakharov, *Generative Models for Fast Calorimeter Simulation: the LHCb case*, *EPJ Web Conf.* **214** (2019) 02034 [1812.01319].

[25] M. Erdmann, J. Glombitza and T. Quast, *Precise simulation of electromagnetic calorimeter showers using a Wasserstein Generative Adversarial Network*, *Comput. Softw. Big Sci.* **3** (2019) 4 [1807.01954].

[26] 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*, *JHEP* **08** (2020) 110 [1903.02433].

[27] E. Buhmann, S. Diefenbacher, E. Eren, F. Gaede, G. Kasieczka, A. Korol et al., *Getting High: High Fidelity Simulation of High Granularity Calorimeters with High Speed*, 2005.05334.

[28] T.I. Collaboration, *International large detector: Interim design report*, 2020.

[29] G.R. Khattak, S. Vallecorsa, F. Carminati and G.M. Khan, *High energy physics calorimeter detector simulation using generative adversarial networks with domain related constraints*, *IEEE Access* **9** (2021) 108899.

[30] CERN, "Welcome to the Compact Linear Collider Website | clic-study.web.cern.ch."

[31] GEANT4 collaboration, *GEANT4: A Simulation toolkit*, *Nucl. Instrum. Meth.* **A506** (2003) 250.

[32] P. Lebrun, L. Linssen, A. Lucaci-Timoce, D. Schulte, F. Simon, S. Stapnes et al., *The CLIC Programme: Towards a Staged e+e- Linear Collider Exploring the Terascale : CLIC Conceptual Design Report*, 1209.2543.

[33] A. Odena, C. Olah and J. Shlens, *Conditional image synthesis with auxiliary classifier GANs*, in *Proceedings of the 34th International Conference on Machine Learning*, D. Precup and Y.W. Teh, eds., vol. 70 of *Proceedings of Machine Learning Research*, (International Convention Centre, Sydney, Australia), pp. 2642–2651, PMLR, 06–11 Aug, 2017, http://proceedings.mlr.press/v70/odena17a.html.

[34] F. Chollet et al., "Keras." https://github.com/fchollet/keras, 2015.

[35] M. Abadi et al., *TensorFlow: Large-scale machine learning on heterogeneous systems*, 2015.

[36] S. Ioffe and C. Szegedy, *Batch normalization: Accelerating deep network training by reducing internal covariate shift*, in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, p. 448–456, JMLR.org, 2015.

[37] A.L. Maas, A.Y. Hannun and A.Y. Ng, *Rectifier nonlinearities improve neural network*

acoustic models, in *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

[38] V. Nair and G.E. Hinton, *Rectified linear units improve restricted boltzmann machines*, in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, (USA), pp. 807–814, Omnipress, 2010, http://dl.acm.org/citation.cfm?id=3104322.3104425.

[39] N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, *Dropout: a simple way to prevent neural networks from overfitting.*, *Journal of Machine Learning Research* **15** (2014) 1929.

[40] G. Hinton, N. Srivastava and K. Swersky, "Lecture 6a overview of mini–batch gradi-ent descent.." 2012.

[41] Q. Xu, G. Huang, Y. Yuan, C. Guo, Y. Sun, F. Wu et al., *An empirical study on evaluation metrics of generative adversarial networks*, 2018.

[42] J. Xie, B. Xu and C. Zhang, *Horizontal and vertical ensemble with deep representation for classification*, *CoRR* **abs/1306.2759** (2013) [1306.2759].

[43] C.W. Fabjan and D. Fournier, *Calorimetry*, in *Particle Physics Reference Library: Volume 2: Detectors for Particles and Radiation*, C.W. Fabjan and H. Schopper, eds., (Cham), pp. 201–280, Springer International Publishing (2020), DOI.

[44] D. Bortoletto, "Detectors for particle physics." URL: https://indico.cern.ch/event/318531/attachments/612850/843143/daniela_l5.pdf.

[45] K. Shmelkov, C. Schmid and K. Alahari, *How good is my GAN?*, *arXiv e-prints* (2018) arXiv:1807.09499 [1807.09499].

[46] S. Vallecorsa, D. Moise, F. Carminati and G.R. Khattak, *Data-parallel training of generative adversarial networks on hpc systems for hep simulations*, in *2018 IEEE 25th International Conference on High Performance Computing (HiPC)*, pp. 162–171, 2018, DOI.

[47] F. Carminati et al., *Generative Adversarial Networks for Fast Simulation: distributed training and generalisation*, in *Proceedings of Artificial Intelligence for Science, Industry and Society — PoS(AISIS2019)*, vol. 372, p. 012, 2020, DOI.