HGCAL Fast Simulation with Deep Learning

AUGUST 2018

AUTHOR:

Vitória Barin Pacela

E4/EP-CMG-PS

SUPERVISORS(S):

Shah Rukh Qasim
Jan Kieseler
Maurizio Pierini

CERN openlab

Project Specification

Highly granular calorimeters (HGCAL) will be one of the biggest novelties of the CMS Phase II upgrade and, in general, of the next generation of collider experiments. This kind of detectors offer more opportunities but much more complexity. It has a drawback on the execution time of generic tasks, such particle reconstruction and identification as well as, notably, event simulation. In order to stay within the technical budgets (e.g. computing time) and satisfy the demand for large simulation samples, experiments will have to work on faster and more accurate simulation process. Deep Learning, and in particular generative models, offer an interesting possibility to speed up the simulation technique. Moreover, deep learning solutions are particularly suitable for HGCAL, given the pixelated nature of its active material. This project aims to adapt to HGCAL existing work on GAN for fast simulation.

Abstract

This project uses Wasserstain Generative Adversatial Networks (WGANs) to supply the demand for large simulation samples in the event of the CMS Phase II Upgrade. The distributions of real and generated hits on different detector layers are trained, revealing good quality especially in the production of longitudinal showers. This result provides a baseline for further studies. The performance of the trained models is evaluated using three different metrics, all giving consistent conclusions. The Wasserstein distance was chosen to decide upon the best model.

Contents

List of Figures

List of Tables

1. Introduction

1.1 Motivation

The highly granular calorimeter (HGCAL) will be the greatest novelty of the Compact Muon Solenoid (CMS) Phase II upgrade [3] and of the next generation of collider experiments. After the upgrade, there will be four times more particle interactions in the experiment, introducing a new demand for large and accurate simulation samples, while detector simulations present higher time complexity in the new calorimeter architecture.

Deep Learning, and in particular generative models, offer an interesting possibility to speed up the simulation technique. Generative Adversarial Networks (GANs) [5] have been applied to calorimeter simulation problems in High Energy Physics (HEP): the CaloGAN [9] architecture considered a simplified version of the electromagnetic calorimeter from the ATLAS experiment, reproducing particle shower properties while achieving significant computational speedup. GANs have been explored in three-dimensional calorimeters [2] [8] by treating detector response simulation as an image generation problem, employing three-dimensional convolutions in the model architecture. More recently, Wasserstein GANs (WGANs) have been applied to calorimeter data from CERN's Super Proton Synchrotron test beam [4].

Deep learning solutions are particularly suitable for the CMS High-Granularity Calorimeter (HGCAL), given the pixelated nature of the problem. This project adapts existing work about GANs for fast simulation. Particular attention is devoted to the specific aspects of HGCAL, such as its hexagon-shaped cell geometry.

The repository for this work is available at `https://github.com/vitoriapacela/hgcal_wgan`.

1.2 Data

The dataset consists of Monte Carlo simulations of particle showers generated with Geant4 software. For each generated particle shower, the following information is recorded about the incoming particle: pseudorapidity (η), azimuthal angle (ϕ), and true energy. Reconstructed hits are described by the energy, time, η, ϕ, and layer–the longitudinal direction in which the shower is developed.

The active material of the HGCAL consists of hexagonal silicon sensors that detect particle showers. In this sense, (η, ϕ, layer) coordinates correspond to the center of the hexagons. In order to approach this data as in an image-generation problem, it needs to be converted to a pixelated image format. For such, the energy deposits are mapped from (η, ϕ, layer) coordinates to a 3-D array. A square tiling is fit into the array, so that each pixel of the grid contains a set of energy deposits, whose size depends on the size of the squares. The tiling size is chosen such that there are at most 6 six sensors per pixel, and then the energy deposits in each pixel are summed up. The data shape after preprocessing is (16, 16, 55). The depth of the calorimeter is 55 layers, while the selected window size for the events is 16x16.

As a first attempt to solve the problem, simulated events of electrons without pileup were selected. Moreover, the considered calorimeter ”window” contains deposits coming from only one particle. The events are stored in compressed HDF5 files. The training set contains 155 145 events with energies ranging from 0 to 500 GeV. A higher number of events could improve the

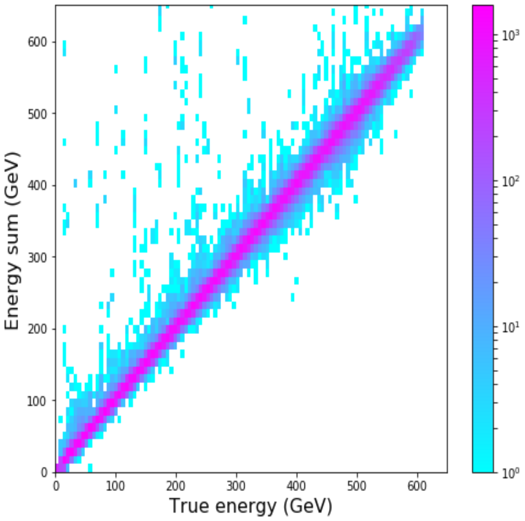

Figure 1.1: The sum over all the pixels of the calorimeter relates to the true energy of the incoming particles.

performance of the model, therefore experiments using larger samples should be performed in the future.

The data describes a small energy loss in the calorimeter – compared to other types of calorimeters, in which the majority of the energy is lost in the absorber –, as illustrated in Fig. 1.1, in which the total energy deposited in the calorimeter has approximately a one-to-one relation to the true energy of the incoming particle. However, in some events the total energy deposited in the calorimeter is higher than the true energy of the particle. This possible problem in the dataset did not have any significant negative impact in training.

The 3D data can be analysed according to its projections in each plane, as illustrated in Fig. 1.2. For each plane projection, the original data is summed into one of the axes for each event, and the image is obtained by taking the average of all the events. There is a slight discrepancy between η bin and ϕ bin axes, clarified in Fig. 1.3. This is explained by the bending of electrons due to the electromagnetic field. It is also noticeable that the electrons decay in the first half of the calorimeter, and that there is a dispersion in the 27^{th} layer of the calorimeter.

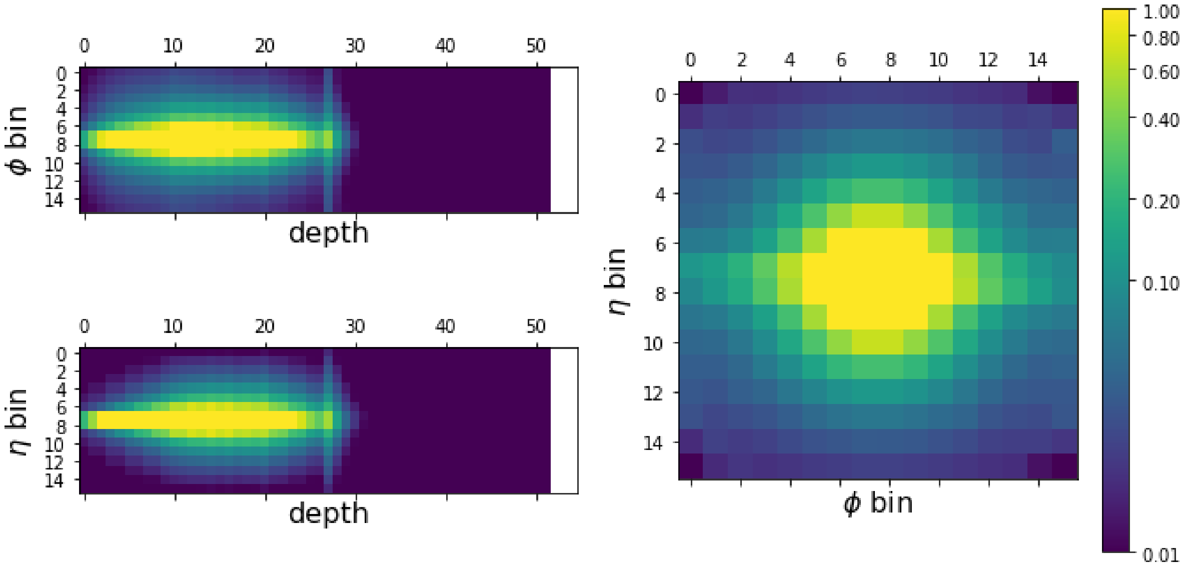

Figure 1.2: Projection of the training data in η x ϕ, η x depth, and ϕ x depth planes.

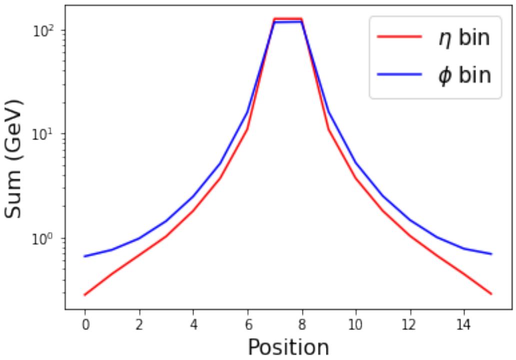

Figure 1.3: Total energy deposited in η and ϕ bins, according to the position. Most of the energy depositions occur in the center of the calorimeter, while the energy depositions over the ϕ bin surpass the depositions over the η bin.

1.3 GANs and WGANs

GANs are generative models that use supervised learning to estimate the cost function. They use adversarial training by defining a generative model (G) that captures the data distribution, and a discriminative model (D) that estimates the probability of whether a sample came from the training data, or from the generative model. The generator is a differentiable function that yields a sample from the generative model given a noise variable z. While D trains, it estimates the ratio between the training data distribution and the model-generated distribution. D can identify the samples that came from the generator because the data lies in a low-dimensional manifold. During training, these two models compete against each other: G should produce samples as close to the training data as possible, in order to maximize the probability of D making a mistake. This is possible through a MiniMax game, in which the loss function (eq. 1.1) should be optimized.

$$\min_{G} \max_{D} L(D,G) = E_{x \sim p_r(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))] \tag{1.1}$$

Even though GANs have been successful in image generation problems, practical training is unstable for various reasons. The largest problem is non-convergence: while each model updates its cost separately, the gradient update in each model may not converge, thus being hard to achieve the Nash equilibrium.

To mitigate such problem, the Wasserstein GAN (WGAN) [1] uses the Wasserstein distance to measure the distance between two probability distributions: from "generated" (generated by the generator model) to "real" (training) data. This measure is also known as the Earth Mover's distance, and it can be interpreted as moving amounts of volume from one distribution to the other at a minimum cost–in terms of quantity of moves and distance–until one distribution turns into the other one. WGANs use the Wasserstein distance as the loss function, according to eq. 1.2.

$$W(p_r, p_g) = \frac{1}{K} \sup_{\|f\|_L \leq K} E_x \sim p_r[f(x)] - E_{x \sim p_g}[f(x)] \tag{1.2}$$

In WGANs the discriminator must lie within the space of 1-Lipschitz functions, which is reinforced with weight clipping: after every gradient update in the discriminator, the weights are limited to a fixed range $[-c, c]$.

Furthermore, the discriminator does not play the role of a critic, but it helps to approximate the Wasserstein metric by trying to maximize the distance between its outputs for real and generated samples. Whereas in the vanilla GAN the discriminator had a sigmoid output, representing the probability that samples are real or generated, in the WGAN the discriminator's output is linear. In addition, it is convenient to label generated samples as -1 and real samples as 1, instead of 0 and 1, so that label multiplication aids the calculation of the loss function.

2. HGCAL WGAN

2.1 Model architecture and training

The model architecture is a variation of the DCGAN (Deep Convolutional GAN) [10]. It uses batch normalization [7] layers with momentum 0.8 both in the discriminator and in the generator, except for the last layer of the generator and first layer of the discriminator, since the model can then learn the mean and scale of the data distribution.

RMSprop [6] is used as the optimizer with learning rate $0.5 \cdot 10^{-4}$, as recommended in the WGAN paper [1], since momentum-based optimizers like Adam cause instability in model training. The models were training using a mini-batch size of 128. The weights were initialized from a uniform distribution with Glorot uniform initializer. The clipping value is 0.05, and the discriminator network is trained five times for every generator training iteration.

Multiple models were tested, with varying types of convoltional layers. This work presents the most successful model in terms of quality of samples generated and training time. The critic network, illustrated in 2.2, processes generated and real images by using two-dimensional convolutional layers, treating the calorimeter layers as image channels of dimension 55, whereas the 16×16 cross section is treated as image rows and columns, respectively. The generator network receives random noise as input as starts from a 100-dimensional latent space. Fractionally-strided convolutions are applied in the same way to generate samples of the same dimensions as the real images: 16x16x55, as shown in Fig. 2.1.

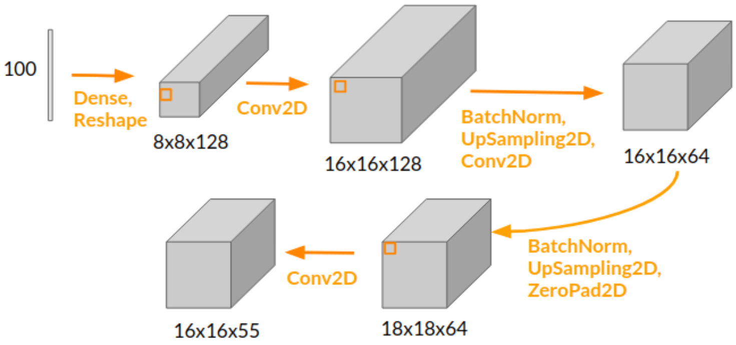

Figure 2.1: Architecture of the generator network. The generator starts from a 100-dimensional uniform distribution projected to a convolutional representation. Three deconvolutions are used until the generated image of the correct shape is reached.

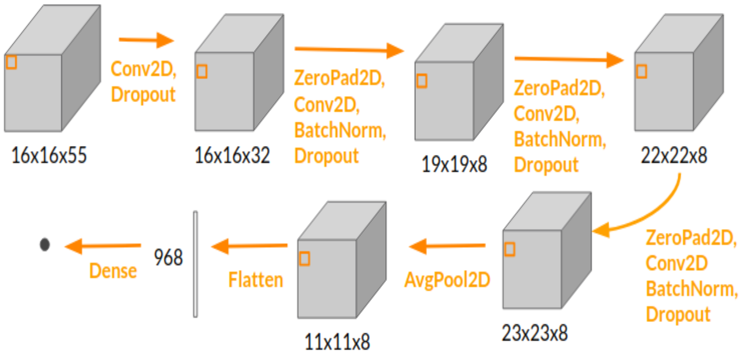
16x16x55
Conv2D, Dropout
16x16x32
ZeroPad2D, Conv2D, BatchNorm, Dropout
19x19x8
ZeroPad2D, Conv2D, BatchNorm, Dropout
22x22x8
ZeroPad2D, Conv2D BatchNorm, Dropout
23x23x8
AvgPool2D
11x11x8
Flatten
968
Dense

Figure 2.2: Architecture of the critic network. Four Convolution2D layers are applied in conjunction with ZeroPadding2D, Batch Normalization, and Dropout layers. The output from the Dense layer does not contain the sigmoid activation function that is usual of discriminator architectures.

2.2 Results

The generated samples are similar to the real ones, but present higher values of energy. The energy deposition in each axis of the calorimeter, shown in Fig. 2.3, reveals that the showers are well-reproduced with respect to the depth of the calorimeter. On the other hand, the generated showers are problematic with respect to the x $-\eta$ bin– axis in its borders. The distribution of energy deposition along the y $-\phi$ bin– axis for generated samples still requires improvement to be better approximated to the distribution of real samples.

Different metrics have been applied for model evaluation: Wasserstein distance, Kullback-Leibler (KL) divergence, and Jensen-Shannon (JS) divergence. They measure the difference between the energy distributions of real and fake samples for each axis (the same distributions shown in Fig. 2.3). The average of the score of each axis is calculated for each metric, and the results are displayed in Fig. 2.4. It is clear that all metrics converge on showing the performance of the model during training; they all indicate that the best model is achieved approximately in step 4000, after which the divergence only increases. It is relevant to point out that the minimum achieved around step 2000, it does not represent a reliable metric due to its uncertainity. Fig. 2.5 shows the Wasserstain distance plotted for each axis. While the distribution along z is learned fast, the distributions along x and y only agree with each other after step 4000.

The training time of the model presented for 2000 steps is of approximately 10 minutes in a V100 Nvidia GPU, as shown in Table 2.1. Furthermore, 1000 samples can be generated in less than a second using 10 CPU cores Skylake.

Time X Machine	Nvidia V100 GPU	Nvidia GeForce GTX 1080	CPUs
Training (2000 steps) (s)	486	928	3373
Generation (1000 samples) (s)	–	0.1	0.69

Table 2.1: Comparison between training and generation time in different machines.

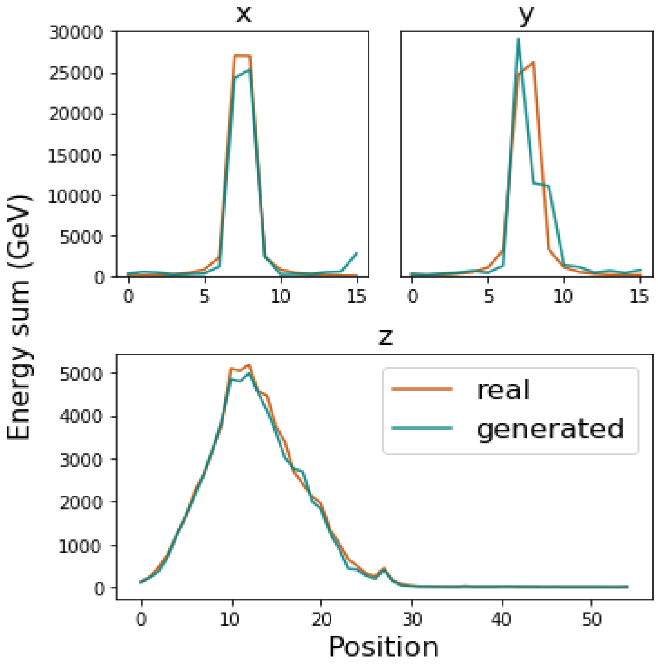

Figure 2.3: Energy deposition per axis. The generated samples demonstrate a similar distribution to the real samples regarding the z axis – depth of the calorimeter. Distributions along x – η bin – and y – ϕ bin – axes require improvement.

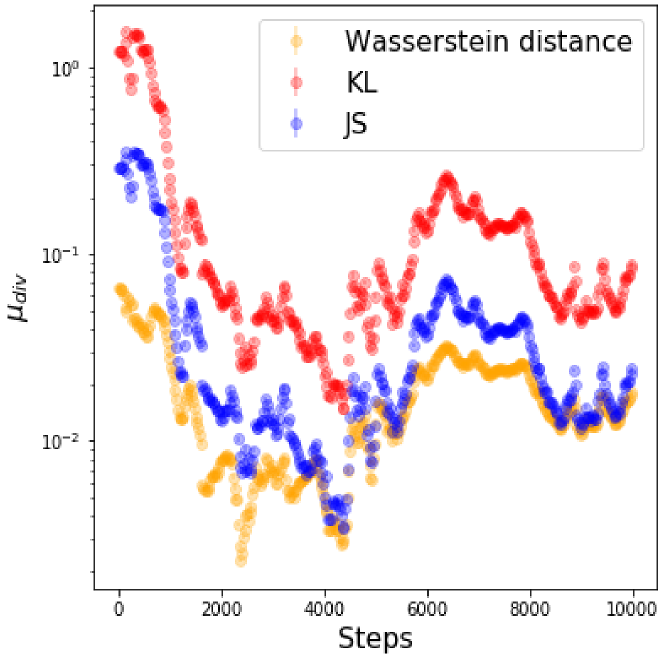

Figure 2.4: Average of the measure of the Wasserstain distance, KL divergence, and JS divergence, calculated on the distributions of energy deposition per axis. The best model is found by step 4000.

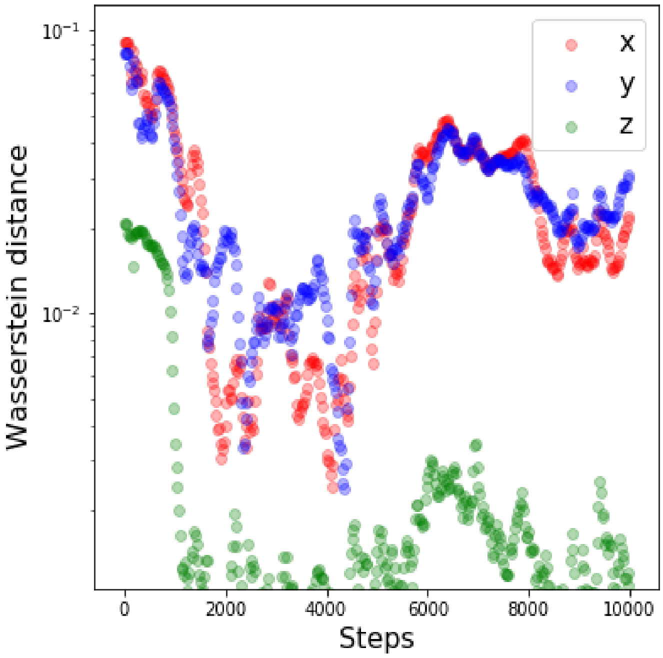

Figure 2.5: Wasserstain distance metric calculated on the distributions of energy deposition per axis. The distribution along the z – depth of the calorimeter– axis is learned fast, while the distribution on x –η bin– and y–ϕ bin– axes stabilize after step 4000.

3. Conclusions

This work proposes the use of WGANs as an alternative to supply the high demand for fast shower simulation for the HGCAL after the CMS Phase II Upgrade, and demonstrates a good baseline that can be further explored. The generated samples have shown to be similar to the real sample distribution, especially in reproducing energy deposits along the depth of the calorimeter. Nevertheless, improvement is still necessary in generating samples in better correspondence to the real distribution over η and ϕ bins. A bigger dataset could auxiliate in this task. Different metrics were employed to evaluate the generated samples, and they all converge both in comparing models and in deciding when to stop training. Comparisons between different models using the Wasserstein difference metric will be presented in the future. In addition to the successful model performance, the proposed framework promises significant simulation speedup: training is performed under 10 minutes in a V100 GPU, whereas the inference time to generate 1000 samples is less than a second in the same machine. Benchmarking in more machines would be interesting. Yet these numbers cannot be directly compared to GEANT yet because the events generated have lower complexity.

For future work, an additional task of energy regression task will be given to the generator, as an attempt to preserve the correlation between the total energy generated per event and how such energy is distributed over the calorimeter, as has been shown in other studies [8]. Constraints on the total energy, η, and ϕ will be imposed to sample generation. Furthermore, other datasets must be explored for different particle types – such as photons and pions – and for events containing pileup.

Zero suppression problems may arise from the simplification of the problem when the data is preprocessed. In a real experiment, sensors will contain inifinitely small energy deposits, which will be different from the generated samples, and can be an identifier to aid the criticial network in discriminating generated from real samples. This issue can be mitigated by inserting a distribution of infinitely small values to the input that is given to the critic. Alternatively, it can be added to the end of the generator network.

After the development of this work it was noticed that the dimension of the depth of the HGCAL is 52 instead of 55. This difference does not affect the results, but should be considered for reimplementing the model architectures to decrease computational time during training.

4. Acknowledgments

I am immensely grateful to my supervisor Maurizio Pierini for all the support offered during this project. I also appreciate the collaboration of Shah-Rukh Qasim and Jan Kieseler in data preprocessing. I would like to thank Sofia Vallecorsa and Gul Rukh Khattak in insightful discussions on the application of GANs for calorimeter simulations, and Jean-Roch Vlimant for instructing me about training of GANs besides supporting me in training GANs under the mpi_learn framework.

This project was received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement no. 772369).

Part of this work was conducted at *"iBanks"*, the AI GPU cluster at Caltech, and part at the VOMS cluster from the Simons Foundation. We acknowledge NVIDIA, SuperMicro and the Kavli Foundation for their support of *"iBanks"*.

Finally, I am thankful to the Openlab organization and fellow summer students for contributing to a wonderful and intense summer.

Bibliography

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *arXiv:1701.07875 [cs, stat]*, January 2017. arXiv: 1701.07875. 4, 5

[2] Federico Carminati, Gulrukh Khattak, Maurizio Pierini, Amir Farbin, Benjamin Hooberman, Wei Wei, Matt Zhang, Vitória Barin Pacela, Soa Vallecorsafac, Maria Spiropulu, and Jean-Roch Vlimant. Calorimetry with Deep Learning: Particle Classication, Energy Regression, and Simulation for High-Energy Physics. page 6, December 2017. 1

[3] CMS Collaboration. The Phase-2 Upgrade of the CMS Endcap Calorimeter. April 2018. Technical Design Report. 1

[4] Martin Erdmann, Jonas Glombitza, and Thorben Quast. Precise simulation of electromagnetic calorimeter showers using a Wasserstein Generative Adversarial Network. *arXiv:1807.01954 [physics]*, July 2018. arXiv: 1807.01954. 1

[5] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. *arXiv:1406.2661 [cs, stat]*, June 2014. arXiv: 1406.2661. 1

[6] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Lecture 6a Overview of mini-batch gradient descent. 2012. 5

[7] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv:1502.03167 [cs]*, February 2015. arXiv: 1502.03167. 5

[8] Gul Rukh Khattak, Sofia Vallecorsa, and Federico Carminati. Three Dimensional Energy Parametrized Generative Adversarial Networks for Electromagnetic Shower Simulation. *ICIP*, 2018. 1, 9

[9] Michela Paganini, Luke de Oliveira, and Benjamin Nachman. Accelerating Science with Generative Adversarial Networks: An Application to 3d Particle Showers in Multi-Layer Calorimeters. *Physical Review Letters*, 120(4), January 2018. arXiv: 1705.02355. 1

[10] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv:1511.06434 [cs]*, November 2015. arXiv: 1511.06434. 5