Published by IOP Publishing for Sissa Medialab

Received: *July 4, 2022*
Revised: *July 27, 2022*
Accepted: *August 22, 2022*
Published: *September 22, 2022*

Calomplification — the power of generative calorimeter models

S. Bieringer,[a,*] A. Butter,[b] S. Diefenbacher,[a] E. Eren,[c] F. Gaede,[c] D. Hundhausen,[a] G. Kasieczka,[a] B. Nachman,[d,e] T. Plehn[b] and M. Trabs[f]

[a]*Institut für Experimentalphysik, Universität Hamburg,*
Luruper Chaussee 149, 22761 Hamburg, Germany
[b]*Institut für Theoretische Physik, Universität Heidelberg,*
Philosophenweg 16, 69120 Heidelberg, Germany
[c]*Deutsches Elektronen-Synchrotron DESY,*
Notkestr. 85, 22607 Hamburg, Germany
[d]*Physics Division, Lawrence Berkeley National Laboratory,*
Berkeley, CA 94720, U.S.A.
[e]*Berkeley Institute for Data Science, University of California,*
Berkeley, CA 94720, U.S.A.
[f]*Department of Mathematics, Karlsruhe Institute of Technology,*
Englerstr. 2, 76131 Karlsruhe, Germany

E-mail: sebastian.guido.bieringer@uni-hamburg.de

Abstract: Motivated by the high computational costs of classical simulations, machine-learned generative models can be extremely useful in particle physics and elsewhere. They become especially attractive when surrogate models can efficiently learn the underlying distribution, such that a generated sample outperforms a training sample of limited size. This kind of GANplification has been observed for simple Gaussian models. We show the same effect for a physics simulation, specifically photon showers in an electromagnetic calorimeter.

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

ArXiv ePrint: 2202.07352

[*] Corresponding author.

https://doi.org/10.1088/1748-0221/17/09/P09028

Contents

2022 JINST 17 P09028

1 Introduction

Particle physics research at colliders is defined by extremely large datasets combined with precision simulations, from first principles all the way to a detailed detector simulation. A reliable generation and simulation chain is crucial to link measurements to fundamental properties of elementary particles. This chain is factorized into two main parts, event generation based on a fundamental Lagrangian and perturbative or non-perturbative quantum field theory, and detector simulations describing the interactions of relativistic particles with the detector. For the upcoming runs of the Large Hadron Collider (LHC), both parts need to gain significantly in speed, to keep up with the size of experimental datasets. One way to achieve this speed gain is to apply modern machine learning (ML) to all levels of the simulation chain. A key tool in this speed-improvement program is deep generative neural networks (NNs) that learn to emulate slower physics-based simulations, replacing the underlying physics by fast and accurate *surrogate models*.

A foundational question with NN surrogate models is, what are the advantages of using the fast simulation compared with the original dataset used for training? Or specifically, how many more events can we sensibly generate from these models before we are limited, for instance, by the training statistics? Without any additional information, we would expect that the statistical power of a generated dataset is at most the same as the dataset used for training. A larger generated sample than the training dataset will then include successively less information per event than the training data, and eventually the information in the generated events will saturate and be dominated by limitations from the network architecture and training. With this pattern in mind [1], we can define an amplification or GANplification factor [2, 3] in terms of an effective sample size for a given surrogate model.

GANplification arises, intuitively, from the fact that neural networks work like classical parametric fits [4, 5], and they are particularly effective when we want to interpolate in many dimensions. This feature is behind the success of the NNPDF parton densities [6] as the first mainstream ML-application in particle theory.

Formally, this fit-like effect is one source of inductive bias, where the underlying assumption is that physics probability densities are smooth. Especially in particle physics, it should be possible to employ other inductive biases, such as symmetries or fundamental invariances in datasets [7–12]. Fast detector simulations benefit from the fact that we can factorize the problem into pieces. Surrogate models are trained to produce a detector response for each outgoing particle. For example, if there is an event with M outgoing particles, each one will be attached to a sampling from the surrogate model. If the training set has N detector interactions, additional combinatorial factors appear for choosing N out of M different events that could be created. These factors can lead to another statistical amplification. Finally, surrogate models with valid inductive biases require far fewer parameters to specify than the original dataset, so there will also be a benefit in the required disk space.

The goal of this paper is to study the statistical amplification of deep generative models, focusing on interpolation from the smoothness inductive bias, for detector simulations as a realistic and highly relevant application. Fast surrogate models for detector simulations have been developed [13–25] and improved [26–40] to the level that they are ready to be used in the upcoming LHC runs. In fact, the ATLAS Collaboration has already integrated a Generative Adversarial Network (GAN) into its fast calorimeter simulation and will use it to generate over a billion events [41, 42]. Initial studies exist on quantifying uncertainties of generative models in event generation [43], but there has not yet been a study of the fundamental benefits of deep generative surrogates applied to detector simulations.

In this paper, we study statistical amplification in the context of photon showers in an electromagnetic calorimeter for a GAN-like generative model (Calomplification). However, the method can be applied to gauge the merit of generative surrogates whenever the underlying distribution can be accessed either through a large number of samples or analytically. We expect similar results in all cases where the smoothness assumption on the underlying density distribution is valid.

The paper is organized as follows. In section 2, we start by introducing our data set and the established generative Variational Autoencoder-GAN (VAE-GAN) architecture adapted to this simulation [30]. Next, we describe our treatment of the comparison between generated and truth samples and the relevant observables in section 3. We then present the amplification effects of the generative networks in section 4. This comparison includes an estimate of the effective sample size to the information encoded and a comparison to standard density estimators. In section 5, we briefly summarize our promising findings.

2 Dataset and model

The International Large Detector (ILD) [44] is one of two detector concepts proposed for the International Linear Collider (ILC). It is optimized towards the Particle Flow analysis concept for optimal global event reconstruction [45, 46]. It combines high-precision tracking and vertexing capabilities with very good hermiticity and highly-granular electromagnetic and hadronic calorimeters (ECal/HCal). We choose one of its two proposed electromagnetic calorimeters, the Si-W ECal, for our dataset. It consists of 30 active silicon layers in a tungsten absorber stack with 20 layers of 2.1 mm and 10 layers of 4.2 mm thickness. The silicon sensors have a cell size of 5×5 mm^2.

ILD uses iLCSoft [47] for detector simulation, reconstruction, and analysis. The Geant4 [48] simulation uses a realistic detector model implemented in DD4hep [49]. Photons are shot into the ECal barrel at a perpendicular incident angle. We project the cells with energy depositions (hits) onto a rectangular grid of $30 \times 30 \times 30$ cells. We choose photon showers, because their structure is more regular and faster to learn than the structure of pion showers [32].

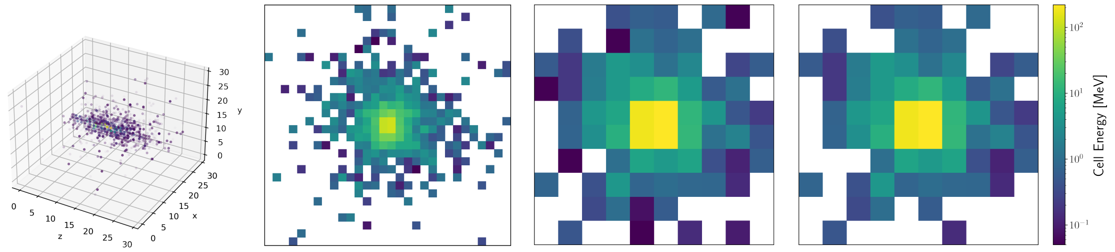


Figure 1. Illustrated transformation of the original calorimeter images from left to right. All histograms feature a logarithmic color coding, with an equal scaling for the 10 × 10 images. The final step of cutting below half the MIP energy is applied for evaluation only.

To develop a high-precision generative model, we fully simulate 268k photon showers with a fixed energy of 50 GeV. From the full set, 50k showers are randomly selected for training (1k) and for the evaluation of the network performance (all 50k). Whenever we need to estimate the generative model uncertainty, we train the network on five sets of 1k training samples. To include an error estimate for the evaluation samples, we use five sets of 5k or 10k evaluation showers, chosen as subsets of these 50k showers. The remaining 218k showers are used as a high-statistics estimate of the truth distribution.

To simplify the training of our precision-generative model, we reduce the dimensionality of the images to 10 × 10 pixels, summing along the beam axis and pooling 3 × 3 patches of the resulting 2D-images. The process is illustrated in figure 1. We will always refer to the combined calorimeter cells as pixels of the calorimeter image. The reduction allows us to obtain a powerful model from a small training set, such that the majority of the data can be used to estimate the truth distribution. Finally, we apply a cut at 0.1 MeV, which corresponds to the most probable energy deposition of a minimal ionizing particle (MIP). Cell energies below have a low signal to noise ratio. To aid the network training, this cut is not present in the training data, but applied on the full set of generated and reference data.

In comparison to studies done in context of proposed, high-granularity tracking calorimeters, these simplifications seem extensive. However, for simulation of the current ATLAS detector the AtlFast3 simulation tool [42] uses 300 individual GANs, each generating only one *eta*-slice of the calorimeter. Each network generates a 2D-image in the radius-*phi*-plane of the detector and only 1000 events are generated to learn the highest energy samples. Albeit, the models are trained including ten-thousands of lower energy samples. We see that in order to facilitate a comparison to a large validation set, the task has not been simplified further than in current applications.

Our generative architecture is a VAE-GAN [50], closely related to the network developed for precision simulations of photon showers [30] and illustrated in figure 2. It closely resembles a standard VAE setup, but deviates in its use of a GAN-like discriminator as a substitute for the usual element-wise reconstruction loss. The loss function is

$$\mathcal{L}_{\text{VAE-GAN}} = \mathcal{L}_{\text{GAN}} + \underbrace{D_{\text{KL}}\left(q_{\text{encoder}}(z|x)|p(z)\right)}_{\mathcal{L}_{\text{prior}}}$$

$$\text{with} \qquad \mathcal{L}_{\text{GAN}} = \mathbb{E}_{x\sim p_{\text{data}}(x)}\Big[\log D(x)\Big] + \mathbb{E}_{z\sim q_{\text{encoder}}(z|x)}\Big[\log(1-D(G(z)))\Big]\,, \qquad (2.1)$$

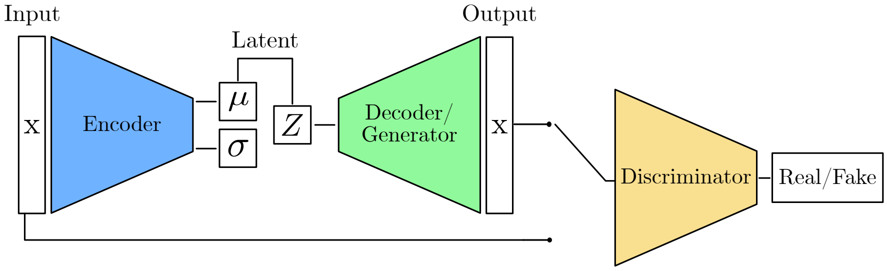


Figure 2. Illustration of the VAE-GAN architecture. The encoder and decoder form a VAE setup, while the decoder can also be understood as a GAN generator. The discriminator acts as a binary classifier, as in a classical GAN.

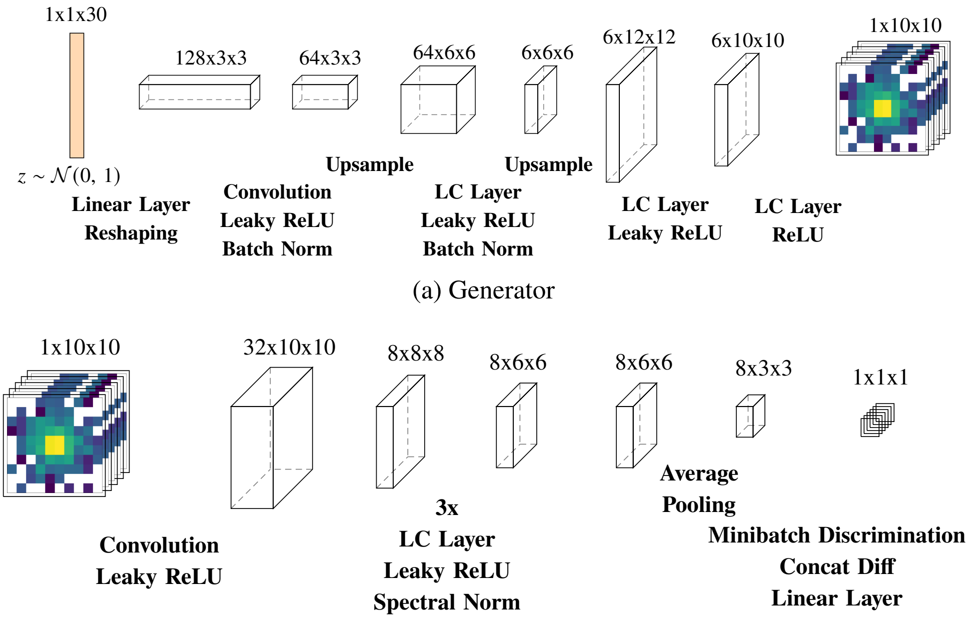
where $p(z) = \mathcal{N}(0, I)$. We maximize $\mathcal{L}_{\text{GAN}}$ during discriminator optimization. Every two discriminator steps, we update generator and encoder by minimizing the full loss function, $\mathcal{L}_{\text{VAE-GAN}}$, i.e. the generator dependent part of $\mathcal{L}_{\text{GAN}}$ and the second term $\mathcal{L}_{\text{prior}}$. The prior loss regularizes the latent space and allows us to sample from $p(z) = \mathcal{N}(0, I)$ during generation. For generator updates, we recast the GAN loss to $-\log D(G(z))$ to ensure efficient training for early epochs [51]. In every update step we sample z only once per input x. Using a GAN-like discriminator is essential, as the range of pixel values covers multiple orders of magnitude. For such images, the element-wise reconstruction loss is dominated by the central, high-energy pixels.

The GAN-like part of our network is modeled after the LAGAN [13] illustrated in figure 3. Unlike many standard applications of convolutional networks, the LAGAN features locally connected layers. Other than convolutional layers, these have the flexibility to account for the missing translation symmetry in calorimeter images. A few changes are made to the original LAGAN setup, including modifying the dimensionality of our network layers to conform to our image and latent space sizes. We also replace batch norm by spectral norm in the discriminator [52] to further stabilize the training. The discriminator uses the difference between reconstructed images and the corresponding training images as an additional input to the final, fully connected layer. For the training images themselves, this difference vector is zero. We apply label smoothing to prevent vanishing gradients from an overconfident classifier. Supplementing the information gained from the images themselves with locally connected layers and mini-batch discrimination [53] ensures better consistency between training and generated images.

The encoder network uses a convolutional input and two convolutional hidden layers, applying Leaky Rectified Linear Unit activation (LeakyReLU) [54] after the first two and ReLU after the third layer. The output of the encoder's convolutional part is fed to two separate linear layers, defining the mean and log var values of the Gaussian VAE latent space.

Our network is implemented in PyTorch 1.8.0 [55] and trained on Nvidia P100s using the `Adam` optimizer [56] with a learning rate of 8×10^{-6} for all networks. Each training on 1k showers is run for 24h, amounting to around 50000 epochs. For epochs after 40000, the distributions of the (i) pixel energy sum (visible energy), (ii) highest pixel energy (peak energy), (iii) per-pixel energy, and, (iv,v) the pixel position weighted by the pixel energy (center of gravity) in a given direction,

$$\left\{ E_{\text{vis}}, E_{\text{peak}}, E_{\text{pixel}}, \text{CG}_{x,y} \right\}, \tag{2.2}$$



(a) Generator

(b) Discriminator

Figure 3. Generator and discriminator setup including parameter space sizes in between operations. The feed-forward for both networks proceeds left to right.

are estimated using histograms of 96000 generated images. The histograms feature 100 bins and constant ranges. Finally, we select the epoch with the best agreement between the generated and training distributions averaged over all five observables, in terms of the measure discussed in the next section. This procedure is repeated for three independent trainings per set of training samples, and we draw a VAE-GAN sample in equal proportions from the resulting three models. We are aware that three independently trained models are not statistically sufficient to define a reliable standard deviation, but we have found them to be very helpful and sufficient in estimating the stability of the network training. The results in section 4 feature the standard deviation on the five different training sets. Whenever we show E_{pixel} we apply an additional minimum cut of 5 MeV, as will be discussed in detail in the next section.

3 Sample comparison

To determine the performance of the trained model, we again use distributions of the same five high-level observables as for the training. We compare showers generated by Geant4 and our VAE-GAN, but now using the high-statistics validation set. Figure 4 shows a set of distributions for 1k shower images used for a single VAE-GAN training and 1000k showers from the corresponding generative network. They are compared to the validation set of 218k Geant4 showers. In addition to the continuous distributions we also show the number of active pixels per image. First, we see that statistical fluctuations of the training set propagate into under- and over-densities of the learned distributions. One prominent difference is the number of active pixels, which can be attributed to the under-estimation of the number of low energy hits below 5 MeV. The remaining learned distributions

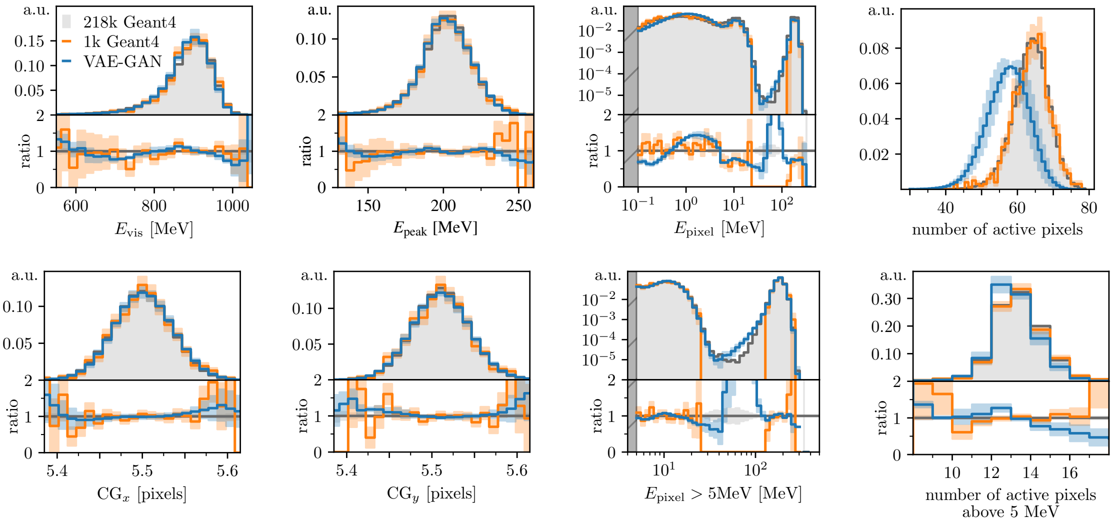


Figure 4. Differential distributions for the observables given in eq. (2.2) from Geant4 and from the VAE-GAN-generated images. Errors of the validation set (grey) and the training set (orange) correspond to the Poisson-error per bin, while the uncertainty on the VAE-GAN line (blue) is illustrated by the standard deviation of three independent trainings on the 1k training data. All histograms are normalized, such that all bins add up to one. The insets show the ratio to the high-statistics estimate of the truth distribution.

are smoother and show fewer fluctuations than the training data. For the visible per-pixel energy, the VAE-GAN interpolates into the sparsely populated interval between around 2 and 120 MeV even though the training set does not include a single pixel in this range. Previous work has shown [30] how to correct the low-energy behavior through an additional, consecutively trained post-processing network, using an maximum mean discrepancy loss [18, 57] on the pixel energy spectrum. Here we skip this post-processing and instead focus on the statistical properties of the generated data for visible pixel energies above 5 MeV.

Quantiles. We now turn to quantifying the efficacy of the VAE-GAN, given the strong performance shown in figure 4. Like in section 2, we could use standard histograms with bins of equal size. However, in this case the occupation numbers of the bins strongly depend on the assumed support of the distributions and on the binning. To avoid zero bins and sparse distributions we have to define the ranges and binnings by hand, making this strategy inconsistent in evaluation. Instead, we now split the support of the distributions into bins of equal probability weight, so-called quantiles, forming the set **Q**. We generate the quantiles for a given distribution by iteratively dividing the set of validation showers into equal-sized subsets and keeping the median as the edge of the quantile. For multi-dimensional distributions, the splitting dimensions alternate. Figure 5 illustrates this algorithm. When comparing generated with reference samples, we want to increase the number of quantiles as far as possible, to cover the entire respective distribution at sufficient resolution.

In this iterative quantile scheme, zero bins will still occur once the number of quantiles exceeds the number of generated showers. To ensure the statistical fluctuations per bin are small and do not cause empty quantiles, we discard results for more than $n/10$ bins, where n is the number of showers in the evaluation set. This leads to roughly 10 events per bin, because the evaluated data is either

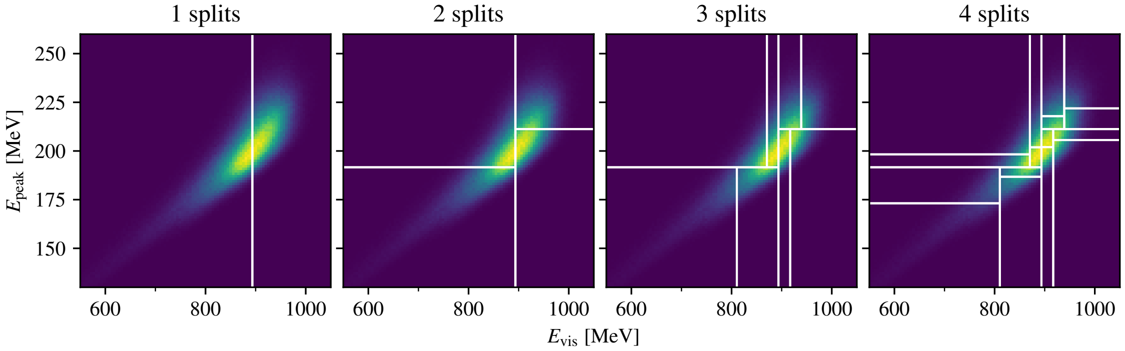


Figure 5. Quantiles developing by splitting of the validation set into subsets of equal size regarding their energy sum & peak energy.

generated from the same distribution as the validation data or is trained to resemble it well. As the event counts follow a Poisson distribution, the probability for a zero bin to occur can be calculated for the average occupation and gives around $4.5 \cdot 10^{-5}$.

Jensen-Shannon divergence. The evaluation chain for the quality of the generated samples starts by constructing quantiles from the validation set. This defines our approximate truth density p_i per quantile i. Next, we extract the density of showers g_i per quantile, either for smaller sets of Geant4 showers or 1000k VAE-GAN generated showers. Due to values appearing in our validation set multiple times, quantiles are not uniquely defined, so the p_i values may differ slightly from their constructed value $1/\#\mathbf{Q}$.

To measure the similarity of the two distributions, we use the Jensen-Shannon divergence

$$D_{\text{JS}}(g,p) = \frac{1}{2} D_{\text{KL}}\left(g \,\middle|\, \frac{g+p}{2}\right) + \frac{1}{2} D_{\text{KL}}\left(p \,\middle|\, \frac{g+p}{2}\right). \tag{3.1}$$

The D_{JS} can be understood as a symmetrized version of the Kullback-Leibler (KL)-divergence

$$D_{\text{KL}}(g\,|\,p) = \int g(x) \log \frac{g(x)}{p(x)} \, \mathrm{d}x. \tag{3.2}$$

For the VAE-GAN results, where $g = g(x)$ is the generated distribution, the D_{JS} is the exact entity optimized by the min-max training on the GAN loss defined in eq. (2.1) [30, 51]. For GAN and Monte Carlo methods, we usually do not have an explicit form of the generated distributions, but only sets $\mathbf{G}$ and $\mathbf{P}$ generated from the estimated distribution g or the true distribution p. This is why we estimate the D_{JS} for the continuous distributions from the quantile values

$$\overline{D}_{\text{JS}}(g,p) = \frac{1}{2} \sum_{Q_i \in \mathbf{Q}} \left(g_i \log \frac{g_i}{\frac{1}{2}(g_i + p_i)} + p_i \log \frac{p_i}{\frac{1}{2}(g_i + p_i)} \right). \tag{3.3}$$

Just like the D_{JS}, this estimate lies between zero and log 2. It turns into the continuous D_{JS} between the histogram estimators

$$\begin{aligned} \overline{g}(x) &= \sum_{Q_i \in \mathbf{Q}} \frac{g_i}{\mathrm{vol}(Q_i)}\, 1_{Q_i}(x) = \sum_{Q_i \in \mathbf{Q}} \frac{\#\{x' \in Q_i \,|\, x' \in \mathbf{G}\}}{\#\mathbf{G} \cdot \mathrm{vol}(Q_i)}\, 1_{Q_i}(x) \\ \text{and} \qquad \overline{p}(x) &= \sum_{Q_i \in \mathbf{Q}} \frac{p_i}{\mathrm{vol}(Q_i)}\, 1_{Q_i}(x)\,, \end{aligned} \tag{3.4}$$

with vol the n-dimensional volume, 1_{Q_i} the indicator function of the i-th quantile and $\mathbf{G}$ all showers in either an evaluation set of Geant4 samples or in the generated set. As for all histogram estimators, independent of the choice of bin edges, the overall number of bins, the cardinality of the fitted set, as well as the number of showers per bin have to go to infinity for the estimator to converge to the underlying distribution. As $\overline{D}_{\mathrm{JS}}$ goes to zero, the two distributions g and p are identical.

To determine the quality of our generative model relative to truth or validation distributions, we look at the dependence of the Jensen-Shannon divergence $\overline{D}_{\mathrm{JS}}$ on the number of quantiles n_{quant} we can reliably construct. This will allow us to gauge where the density estimation underlying the VAE-GAN beats the statistically limited training data. As discussed earlier, we estimate the uncertainty on $\overline{D}_{\mathrm{JS}}$ for the 5k and 10k evaluation sets of Geant4 data from five independent sets each.

4 GANplification performance

Using our illustrated methodology we are now in a position to extend the toy study of ref. [1] to a relevant physics application, with the corresponding increased complexity and physics content.

Overcoming training statistics. In figure 6 we show how $\overline{D}_{\mathrm{JS}}$ depends on the number of quantiles for the different observables given in eq. (2.2). For simple, uni-modal distributions like the energy sum, the peak energy and the centers of gravity, 1000k showers generated from the VAE-GAN achieve similar values as the 1k training data for very low numbers of bins. This means the generated data closely resembles the mean, standard deviation and low-level moments of the training data. For the more complex distribution of the visible per-pixel energy, the $\overline{D}_{\mathrm{JS}}$ only resolves part of the high-density regions for a small number of quantiles. Increasing the numbers of quantiles, the interpolation of the generative model in the sparsely populated areas of the support starts to becomes apparent, and the $\overline{D}_{\mathrm{JS}}$-values for the Geant4 data increases over the VAE-GAN level. As there are on average about 13 active pixels above 5 MeV, as seen in figure 4, the statistics for the per-pixel energy distribution benefits from these 13 pixel measurements per shower. For large numbers of quantiles, the $\overline{D}_{\mathrm{JS}}$ values of the VAE-GAN are consistently below the corresponding values for the training sample and for all observables. This amplification is a result of the interpolation via the generative model's smoothing properties.

To quantify the amplification, we can compare the VAE-GAN distributions to larger Geant4 samples. Again, for small numbers of quantiles the VAE-GAN does not reach the truth $\overline{D}_{\mathrm{JS}}$-values of larger data samples. This confirms that the neural network does not add global information to the training data and will not improve, for instance, the estimated mean of a Gaussian distribution. On the other hand, what we are really interested in are the features over the full distributions. In figure 6 we show how the network trained on 1k showers and used to generate 1000k showers plateaus in $\overline{D}_{\mathrm{JS}}$, as a function of the resolution, and how this plateau value compares to different Geant4 sample sizes. For a large number of quantiles and probing detailed features of the distributions, our VAE-GAN surrogate description corresponds to at least 50k Geant4 showers when we look at E_{vis}, E_{peak}, or

E_{pixel}. This gives us GANplification factors as large as 50 for the relevant high-resolution features. For the reconstructed center of gravity this factor becomes a little smaller, but remains above ten.

Similar observations can be made for joint distributions, or correlations, of the different observables. Figure 7 shows how the VAE-GAN encodes the correlations between observables with a consistently smaller error than the training data. The per-pixel energy distribution cannot be included in the correlations, as it features a varying number of pixel energy values per shower, whereas all other observables give a single value per shower. An unexpected upwards slope appears when examining joint distributions containing the energy sum and the peak energy of the generated images. This can be traced back to slight, small-scale fluctuations in the correlation between them in the generated data. Still, in all of the correlations we find a GANplification factor larger than 50 for the relevant detailed features, larger than for the one-dimensional distributions, as expected from the higher dimensionality and therefore reduced per-quantile statistics.

Density estimation. After we have seen that it is beneficial to generate datasets based on a learned density estimation, the question is whether other ways to estimate densities can give similar results. While there exists literature on convergence rates of generative methods [58, 59], our physics application is defined by very specific limitations, different from those formal arguments. We therefore compare the performance of our VAE-GAN to two classical density estimation techniques. For both of them we analyze the same one-dimensional and multi-dimensional kinematic distributions as before.

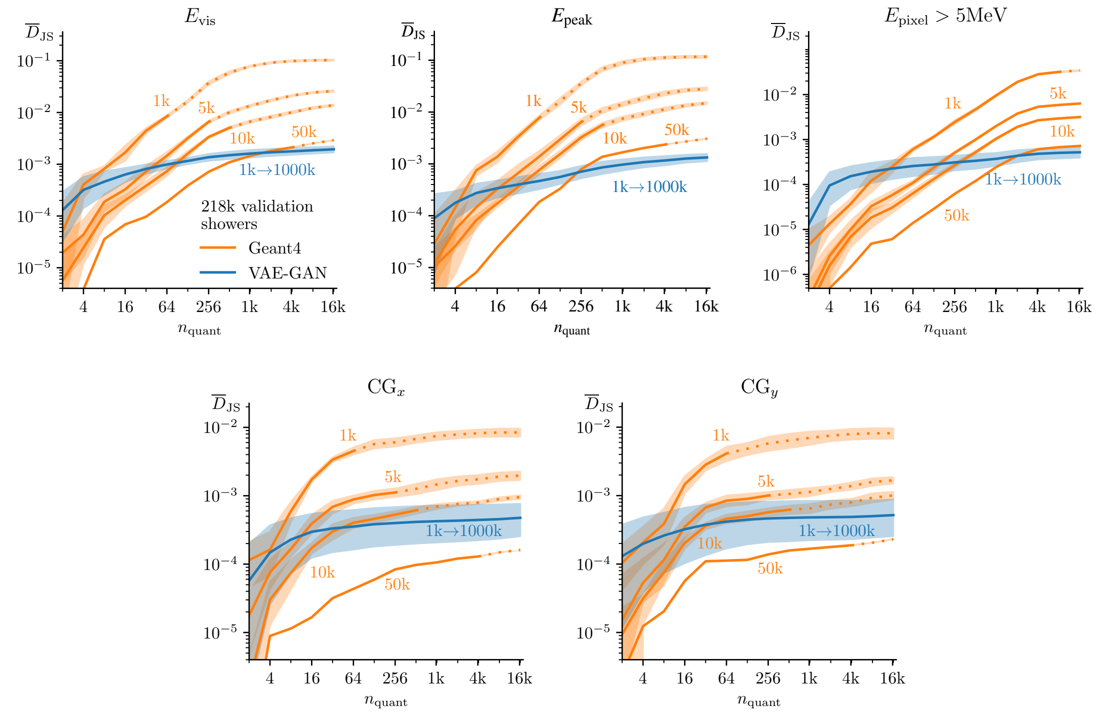


Figure 6. Dependence of $\overline{D}_{\text{JS}}$ on the number of quantiles n_{quant} for different amounts of Geant4 data (orange) and VAE-GAN data (blue) for the observables given in eq. (2.2). Solid lines indicate meaningful, non-sparse quantile sets. The 1k Geant4 samples were also used to train the VAE-GAN. Errors are calculated as the standard deviation from five datasets. For 50k we omit the negligible errors.

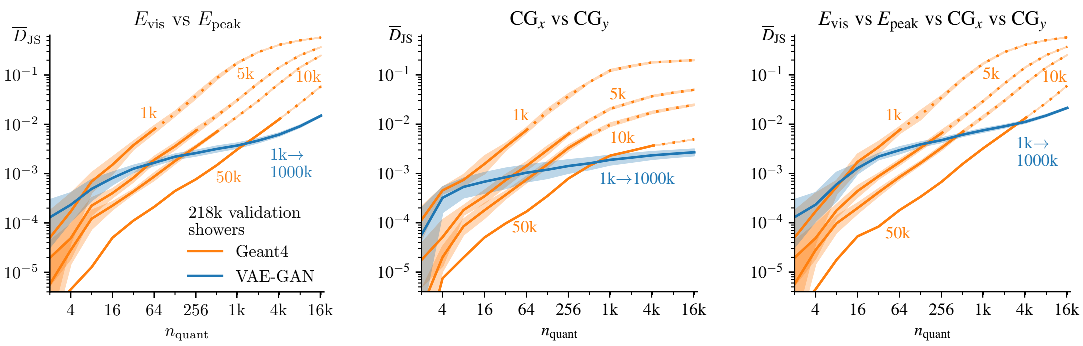


Figure 7. Dependence of $\overline{D}_{\mathrm{JS}}$ on the number of quantiles n_{quant} for different amounts of Geant4 data (orange) and VAE-GAN data (blue), now for correlations between the observables of eq. (2.2), corresponding to the 1D results in figure 6.

To each of our five training sets, we fit a kernel density estimator (KDE) and a histogram estimator, by minimizing the mean negative log-likelihood of cross-validation subsets of the training set on a grid of the estimator parameters. The results for the energy sum are shown in figure 8. For the KDE we use the `scikit-learn` [60] KernelDensity class together with the built-in GridSearchCV tool using 5-fold cross-validation to optimize the bandwidth of the Gaussian kernel. The values of the bandwidth for the individual optimizations are given in table 1. The parameters of the histogram estimator, i.e. the number of bins along the individual dimensions, are optimized using our own implementation of the same techniques. To ensure stable convergence, we form 500 cross-validation sets from the training data. The results of this optimization can again be found in table 1.

In figure 8 we see that the KDE tends to over-fit and that the histogram estimator is limited by its discrete functional form. We can analyze their performance more quantitatively using the $\overline{D}_{\mathrm{JS}}$ shown in figure 9. Due to the logarithmic nature and complex functional form of the per-pixel energy distribution, the histogram estimator and the KDE do not converge for the low number of training showers we use, so we omit this observable. First, trained on 1k showers, the histogram estimator can only use very few bins to balance over-fitting against the approximation error caused by its coarse structure and is thus outperformed by the KDE. For a larger training set and the

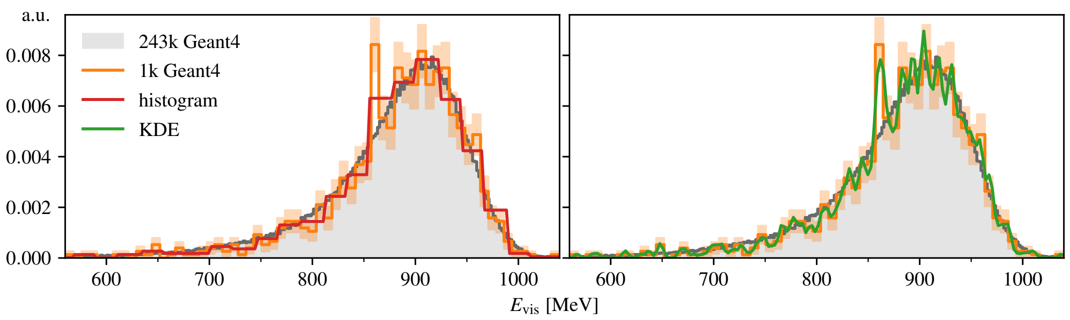


Figure 8. Example of an histogram estimator (red) and a kernel density estimator (green). The orange histogram shows the training data, including Poisson errors, that both estimators where fitted to using cross-validation.

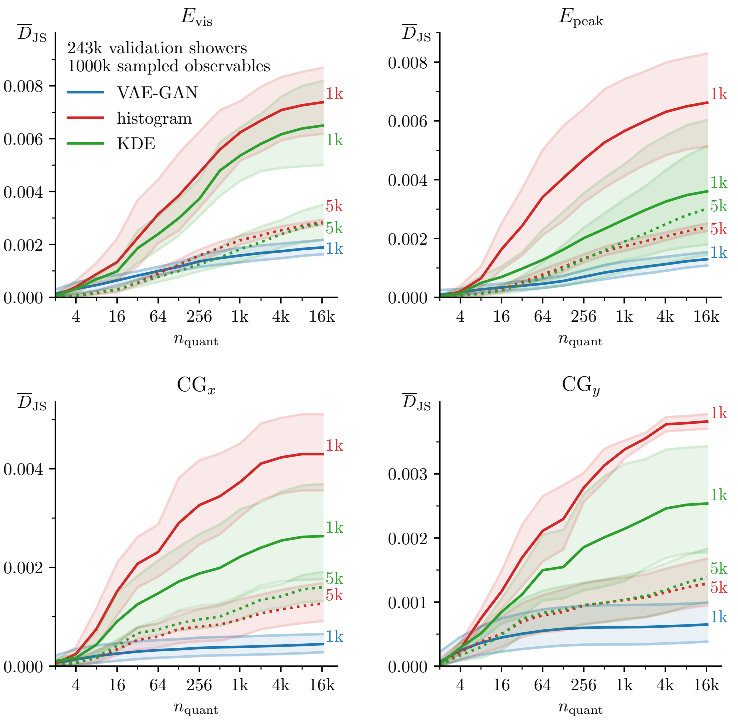


Figure 9. Dependence of $\overline{D}_{\mathrm{JS}}$ on the number of quantiles n_{quant} for 1000k observable values sampled from histogram estimators (red) and kernel density estimators (green) and for 1000k showers sampled from VAE-GANs (blue). Errors are calculated as the standard deviation of five fits to different datasets. The size of the training sets is given to the right of the corresponding lines.

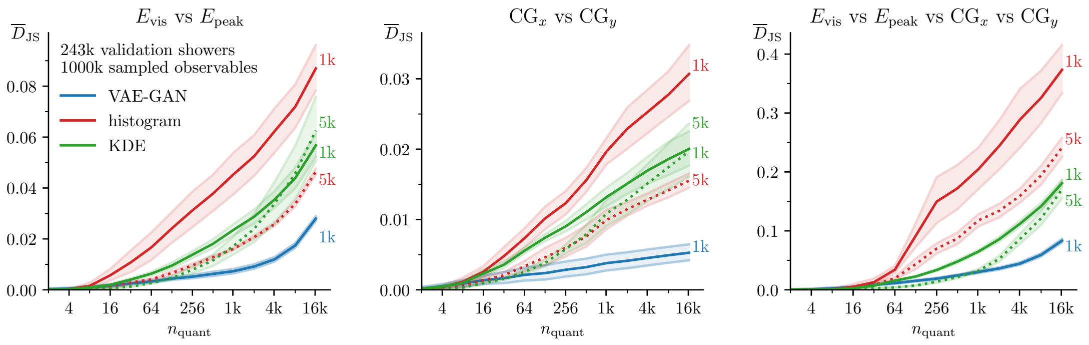


Figure 10. Dependence of $\overline{D}_{\mathrm{JS}}$ on the number of quantiles n_{quant} for 1000k observable values sampled from different density estimators for multi-dimensional combinations of the observables given in eq. (2.2), in analogy to the 1D results in figure 9.

correspondingly larger number of bins, the approximation errors drop and both estimation methods perform similarly. However, compared to the VAE-GAN, both techniques lack descriptive power for small scales. Only for two to four bins they perform similarly to the VAE-GAN. Next, comparing the generative network to density estimators fitted to 5k showers, we can again observe the benefits of higher statistics for estimating low moments of the distributions.

For the 2-dimensional correlations shown in figure 10 we find similar limitations of the classical

Table 1. KDE bandwidths and numbers of bins in the according dimensions for the histogram estimators presented in figures 9 and 10. Estimators are fitted for five independent training sets to extract the mean and standard deviation.

	KDE bandwidth 1k	KDE bandwidth 5k	# histogram bins 1k	# histogram bins 5k
E_{vis}	0.05 ± 0.01	0.03 ± 0.01	39 ± 10	58 ± 7
E_{peak}	0.10 ± 0.03	0.03 ± 0.02	27 ± 4	50 ± 5
CG_x	0.10 ± 0.02	0.02 ± 0.01	32 ± 12	49 ± 13
CG_y	0.10 ± 0.02	0.03 ± 0.01	25 ± 4	43 ± 7
E_{vis} vs E_{peak}	0.09 ± 0.01	0.03 ± 0.01	30 ± 3 × 26 ± 7	40 ± 2 × 46 ± 5
CG_x vs CG_y	0.18 ± 0.02	0.07 ± 0.01	21 ± 1 × 20 ± 1	21 ± 1 × 22 ± 2
complete 4D	0.24 ± 0.01	0.12 ± 0.01	20 × 19 × 5 × 5 ± 1	21 × 21 × 7 × 8 ± 1

methods. Only the 4-dimensional density estimation behaves differently in that the histogram estimator is generally outperformed by the KDE. We can understand these patterns from the histogram parameters in table 1. As the histogram estimator introduces bins in every direction, the number of showers per bin drops inversely proportional to the volume of the space. To avoid over-fitting, only few bins per dimension can then be used, leading to a large approximation error. The KDE and the VAE-GAN scale better with the number of dimensions, and as before the KDE only matches the VAE-GAN performance for a very small number of quantiles.

In addition to the neural network outperforming both density estimators, we remind ourselves that the VAE-GAN actually performs the more general task of estimating the distribution of calorimeter images or low-level observables, whereas the classical methods estimate the distributions of the high-level observables.

5 Conclusions

In this paper we have shown that a realistic generative ML-model can indeed be used to generate a large number of showers, beyond a limited training statistics. Specifically, we used a VAE-GAN to generate photon showers for the electromagnetic calorimeter of the planned ILD detector design at a future linear collider. Our model is a simplification of the established precision-simulation network developed for this task [30]. This model is trained on a small number of showers from a Geant4 simulation, where a high-statistics sample of Geant4 showers serves as a truth estimate. Relative to this truth sample, we estimate the information content of finite-size samples using quantiles for standard kinematic observables and their correlations. A variable number of quantiles allows us to balance resolution with statistics.

Our study confirms earlier results based on a simple Gaussian example [1], in that for a properly trained network a set of generated showers comparable in size to the training data provides a physics-wise nearly equivalent but statistically independent copy of the training data. More generated showers will, individually, contain less information than an actual shower, but add information as a sample. This amount of information can be linked to an effective sample size of actual data. For very large numbers of generated showers, the information in the generated sample reaches a plateau, reflecting limitations of the network architecture and training.

For our problem and network at hand, we find that the effective sample sizes give an enhancement or GANplification factor of 10 to 50, for a large number of quantiles and corresponding to high-resolution kinematic features. For a training sample of 1k showers we generate up to 1000k showers from the network and find a comparable performance of up to 50k Geant4 showers for the kinematic distributions and their correlations. We also interpret the VAE-GAN as a density estimator and find that it learns the truth density from the showers better than standard density estimators on the high-level kinematic variables. This proves that the generative network can even learn and sample from implicitly defined distributions and benefit from superior interpolation or fit properties. These properties motivate deep generative detector simulations for statistical amplification in addition to computational acceleration.

Acknowledgments

We would like to thank Suada Mulgeci for the valuable discussions in the earlier stages of this project. The research of AB and TP is supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grant 396021762 — TRR 257 Particle Physics Phenomenology after the Higgs Discovery. This work was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC 2181/1 — 390900948 (the Heidelberg STRUCTURES Excellence Cluster). BN is supported by the U.S. Department of Energy, Office of Science under contract DE-AC02- 05CH11231. SB is supported by the Helmholtz Information and Data Science Schools via DASHH (Data Science in Hamburg — HELMHOLTZ Graduate School for the Structure of Matter) with the grant HIDSS-0002. DH and SD acknowledge support by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy — EXC 2121 "Quantum Universe" — 390833306. EE is funded through the Helmholtz Innovation Pool project ACCLAIM that provided a stimulating scientific environment for parts of the research done here. This research was supported in part through the Maxwell computational resources operated at Deutsches Elektronen-Synchrotron DESY, Hamburg, Germany.

References

[1] A. Butter, S. Diefenbacher, G. Kasieczka, B. Nachman and T. Plehn, *GANplifying event samples*, *SciPost Phys.* **10** (2021) 139 [arXiv:2008.06545].

[2] Y. Hao, A. Orlitsky, A.T. Suresh and Y. Wu, *Data amplification: A unified and competitive approach to property estimation*, arXiv:1904.00070.

[3] B. Axelrod, S. Garg, Y. Han, V. Sharan and G. Valiant, *On the Statistical Complexity of Sample Amplification*, arXiv:2201.04315.

[4] M. Bellagente, M. Haußmann, M. Luchmann and T. Plehn, *Understanding Event-Generation Networks via Uncertainties*, *SciPost Phys.* **13** (2022) 003 [arXiv:2104.04543].

[5] I. Chahrour and J.D. Wells, *Comparing machine learning and interpolation methods for loop-level calculations*, *SciPost Phys.* **12** (2022) 187 [arXiv:2111.14788].

[6] NNPDF collaboration, *Unbiased determination of the proton structure function F(2)**p with faithful uncertainty estimation*, *JHEP* **03** (2005) 080 [hep-ph/0501067].

[7] S. Krippendorf and M. Syvaeri, *Detecting Symmetries with Neural Networks*, arXiv:2003.13679.

[8] G. Barenboim, J. Hirn and V. Sanz, *Symmetry meets AI*, *SciPost Phys.* **11** (2021) 014 [arXiv:2103.06115].

[9] B.M. Dillon, G. Kasieczka, H. Olischlager, T. Plehn, P. Sorrenson and L. Vogel, *Symmetries, safety, and self-supervision*, *SciPost Phys.* **12** (2022) 188 [arXiv:2108.04253].

[10] C.G. Lester, *Chiral Measurements*, arXiv:2111.00623.

[11] R. Tombs and C.G. Lester, *A method to challenge symmetries in data with self-supervised learning*, 2022 *JINST* **17** P08024 [arXiv:2111.05442].

[12] K. Desai, B. Nachman and J. Thaler, *Symmetry discovery with deep learning*, *Phys. Rev. D* **105** (2022) 096031 [arXiv:2112.05722].

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

[14] M. Paganini, L. de Oliveira and B. Nachman, *Accelerating Science with Generative Adversarial Networks: An Application to 3D Particle Showers in Multilayer Calorimeters*, *Phys. Rev. Lett.* **120** (2018) 042003 [arXiv:1705.02355].

[15] 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 [arXiv:1712.10321].

[16] S. Vallecorsa, F. Carminati and G. Khattak, *3D convolutional GAN for fast simulation*, *EPJ Web Conf.* **214** (2019) 02010.

[17] S. Carrazza and F.A. Dreyer, *Lund jet images from generative and cycle-consistent adversarial networks*, *Eur. Phys. J. C* **79** (2019) 979 [arXiv:1909.01359].

[18] A. Butter, T. Plehn and R. Winterhalder, *How to GAN LHC Events*, *SciPost Phys.* **7** (2019) 075 [arXiv:1907.03764].

[19] R. Di Sipio, M. Faucci Giannelli, S. Ketabchi Haghighat and S. Palazzo, *DijetGAN: A Generative-Adversarial Network Approach for the Simulation of QCD Dijet Events at the LHC*, *JHEP* **08** (2019) 110 [arXiv:1903.02433].

[20] 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 [arXiv:1812.01319].

[21] P. Musella and F. Pandolfi, *Fast and Accurate Simulation of Particle Detectors Using Generative Adversarial Networks*, *Comput. Softw. Big Sci.* **2** (2018) 8 [arXiv:1805.00850].

[22] K. Deja, T. Trzcinski and Ł. Graczykowski, *Generative models for fast cluster simulations in the TPC for the ALICE experiment*, *EPJ Web Conf.* **214** (2019) 06003.

[23] L. de Oliveira, M. Paganini and B. Nachman, *Controlling Physical Attributes in GAN-Accelerated Simulation of Electromagnetic Calorimeters*, *J. Phys. Conf. Ser.* **1085** (2018) 042017 [arXiv:1711.08813].

[24] J.W. Monk, *Deep Learning as a Parton Shower*, *JHEP* **12** (2018) 021 [arXiv:1807.03685].

[25] J.N. Howard, S. Mandt, D. Whiteson and Y. Yang, *Learning to simulate high energy particle collisions from unlabeled data*, *Sci. Rep.* **12** (2022) 7567 [arXiv:2101.08944].

[26] M. Erdmann, L. Geiger, J. Glombitza and D. Schmidt, *Generating and refining particle detector simulations using the Wasserstein distance in adversarial networks*, *Comput. Softw. Big Sci.* **2** (2018) 4 [arXiv:1802.03325].

[27] 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 [arXiv:1807.01954].

[28] M. Backes, A. Butter, T. Plehn and R. Winterhalder, *How to GAN Event Unweighting*, *SciPost Phys.* **10** (2021) 089 [arXiv:2012.07873].

[29] D. Belayneh et al., *Calorimetry with deep learning: particle simulation and reconstruction for collider physics*, *Eur. Phys. J. C* **80** (2020) 688 [arXiv:1912.06794].

[30] E. Buhmann et al., *Getting High: High Fidelity Simulation of High Granularity Calorimeters with High Speed*, *Comput. Softw. Big Sci.* **5** (2021) 13 [arXiv:2005.05334].

[31] E. Buhmann et al., *Decoding Photons: Physics in the Latent Space of a BIB-AE Generative Network*, *EPJ Web Conf.* **251** (2021) 03003 [arXiv:2102.12491].

[32] E. Buhmann et al., *Hadrons, better, faster, stronger*, *Mach. Learn. Sci. Tech.* **3** (2022) 025014 [arXiv:2112.09709].

[33] C. Krause and D. Shih, *CaloFlow: Fast and Accurate Generation of Calorimeter Showers with Normalizing Flows*, arXiv:2106.05285.

[34] C. Krause and D. Shih, *CaloFlow II: Even Faster and Still Accurate Generation of Calorimeter Showers with Normalizing Flows*, arXiv:2110.11377.

[35] G.R. Khattak, S. Vallecorsa, F. Carminati and G.M. Khan, *Fast simulation of a high granularity calorimeter by generative adversarial networks*, *Eur. Phys. J. C* **82** (2022) 386 [arXiv:2109.07388].

[36] R. Kansal et al., *Particle Cloud Generation with Message Passing Generative Adversarial Networks*, arXiv:2106.11535.

[37] A. Hariri, D. Dyachkova and S. Gleyzer, *Graph Generative Models for Fast Detector Simulations in High Energy Physics*, arXiv:2104.01725.

[38] F. Rehm et al., *Reduced Precision Strategies for Deep Learning: A High Energy Physics Generative Adversarial Network Use Case*, arXiv:2103.10142.

[39] F. Rehm, S. Vallecorsa, K. Borras and D. Krücker, *Validation of Deep Convolutional Generative Adversarial Networks for High Energy Physics Calorimeter Simulations*, arXiv:2103.13698.

[40] F. Rehm, S. Vallecorsa, K. Borras and D. Krücker, *Physics Validation of Novel Convolutional 2D Architectures for Speeding Up High Energy Physics Simulations*, *EPJ Web Conf.* **251** (2021) 03042 [arXiv:2105.08960].

[41] ATLAS collaboration, *Deep generative models for fast shower simulation in ATLAS*, ATL-SOFT-PUB-2018-001 (2018).

[42] ATLAS collaboration, *AtlFast3: the next generation of fast simulation in ATLAS*, *Comput. Softw. Big Sci.* **6** (2022) 7 [arXiv:2109.02551].

[43] A. Butter et al., *Generative Networks for Precision Enthusiasts*, arXiv:2110.13632.

[44] ILD Concept Group collaboration, *International Large Detector: Interim Design Report*, arXiv:2003.01116.

[45] M.A. Thomson, *Particle Flow Calorimetry and the PandoraPFA Algorithm*, *Nucl. Instrum. Meth. A* **611** (2009) 25 [arXiv:0907.3577].

[46] J.S. Marshall and M.A. Thomson, *The Pandora Software Development Kit for Pattern Recognition*, *Eur. Phys. J. C* **75** (2015) 439 [arXiv:1506.05348].

[47] *iLCSoft Project Page*, https://github.com/iLCSoft (2016).

[48] J. Allison et al., *Recent developments in Geant4*, *Nucl. Instrum. Meth. A* **835** (2016) 186.

[49] M. Frank, F. Gaede, C. Grefe and P. Mato, *DD4hep: A Detector Description Toolkit for High Energy Physics Experiments*, *J. Phys. Conf. Ser.* **513** (2014) 022010.

[50] A.B.L. Larsen, S.K. Sønderby, H. Larochelle and O. Winther, *Autoencoding beyond pixels using a learned similarity metric*, in *Proceedings of the 33rd International Conference on International Conference on Machine Learning — Volume 48*, pp. 1558-1566, PMLR (2016) [PDF].

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

[52] T. Miyato, T. Kataoka, M. Koyama and Y. Yoshida, *Spectral Normalization for Generative Adversarial Networks*, arXiv:1802.05957.

[53] T. Salimans et al., *Improved techniques for training gans*, in *Adv. Neural Inf. Process. Syst.*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon and R. Garnett eds., vol. 29, Curran Associates, Inc. (2016) [PDF].

[54] A.L. Maas, A.Y. Hannun and A.Y. Ng, *Rectifier nonlinearities improve neural network acoustic models*, in *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, Atlanta, Georgia, U.S.A. (2013).

[55] A. Paszke et al., *PyTorch: An Imperative Style, High-Performance Deep Learning Library*, *Adv. Neural Inf. Process. Syst.* **32** (2019) 8024.

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

[57] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf and A. Smola, *A kernel method for the two-sample-problem*, in *Adv. Neural Inf. Process. Syst.*, B. Schölkopf, J. Platt and T. Hoffman eds., vol. 19, MIT Press (2007) [PDF].

[58] G. Biau, B. Cadre, M. Sangnier and U. Tanielian, *Some theoretical properties of gans*, *Annals Statist.* **48** (2018) 1539 [arXiv:1803.07819].

[59] D. Belomestny, E. Moulines, A. Naumov, N. Puchkin and S. Samsonov, *Rates of convergence for density estimation with GANs*, arXiv:2102.00199.

[60] F. Pedregosa et al., *Scikit-learn: Machine learning in Python*, *J. Mach. Learn. Res.* **12** (2011) 2825.