CaloPointFlow - Generating Calorimeter Showers as Point Clouds

Simon Schnake[1,2], **Benno Kach**[1], **Moritz Scham**[1,2], **Dirk Krücker**[2], **Kerstin Borras**[1,2], **Sofia Vallecorsa**[3]

[1] Deutsches Elektronen-Synchrotron DESY, Notkestrasse 85, 22607 Hamburg, Germany
[2] RWTH Aachen University, Templergraben 55, 52062 Aachen, Germany
[3] CERN, 1211 Geneva 23, Switzerland

E-mail: `simon.schnake@desy.de`

Abstract. Precise simulations are essential for advancing particle physics research, but they require high computational resources due to the complexity of calorimeter interactions. To reduce this cost, we propose a novel generative model that interprets calorimeter showers as point clouds rather than 3D images. Each hit is modelled as part of a hit distribution depending on a global latent calorimeter shower distribution. Our model is based on PointFlow (Yang et al., 2019) and consists of a permutation invariant encoder and two normalizing flows: one for modelling the global latent distribution and another for modelling individual hits conditioned on it. We present first results and compare them with state-of-the-art voxel methods.

1. Introduction

Particle physics requires precise simulations to achieve scientific progress. More than half of the computational resources of the Large Hadron Collider (LHC) computing grid are already used for simulation [1]. In the future High Luminosity Phase of the LHC, about 100 times as many simulated events will be needed [2]. The simulation of calorimeters is typically one of the most computationally intensive parts of a detector simulation, and there is an ongoing effort to reduce this computational cost using machine learning.

A high-energetic particle interacting with the material of a calorimeter typically creates a cascade of secondary particles, which in turn interact with the material of the calorimeter, creating an avalanche of lower-energy particles. Modern calorimeters often consist of a sandwich of passive absorber material and active high granular sensors which record the energy deposit (the so-called hits).

In previous attempts at generative modelling of calorimeter showers with Deep Learning, the energy deposits were considered as a 3D grid of voxels and the individual shower as a 3D image. Most models [3–11] used to study the artificial generation of calorimeter data, were *Generative Adversarial Networks* (GANs) [12]. GANs can have relatively flexible architectures and are fast generators, but they suffer from several shortcomings: incomplete distribution coverage (*mode collapse*) and convergence is often difficult to achieve. Recent developments extend the plain GAN approach and combinations of GANs and Autoencoders [13, 14] are used. A completely different approach are Normalizing Flows [15] which allow likelihood-based training also on calorimeter data [16, 17]. This lead to high quality results. One drawback of the CaloFlow [16] approach and normalizing flows in general is that they do not scale well to large dimensions and therefore are not directly applicable for simulating high granular detectors with many active cells, such as the future CMS forward calorimeter HGCal [18] or the proposed CALICE calorimeter [19]. To scale likelihood-training to high dimensions, *Mikuni et. al.* [20] proposed to use score-based diffusion models instead of normalizing flows [15]. We take a different route of scaling likelihood-base training. All previous mentioned models are trained on voxel-based data. This has multiple disadvantages. First, in a high granular calorimeter most cells have no entries, therefore the data is sparse. Second, an irregularly shaped calorimeter is often not directly transferable to voxels. The architecture of the model has to be developed directly for the calorimeter. So one model is not directly applicable for another calorimeter dataset. This leads to worse comparison possibilities of the different models. We interpret the hits of the showers as point clouds. This resolves the sparsity of the data, since empty cells are dropped. Also, the geometry of the calorimeter is transferable, and the model can be used for different datasets without altering the model structure.

2. Model

The model here described is based on *PointFlow* [21]. The target of the PointFlow paper was to model the surface of objects as point clouds. PointFlow used continuous normalizing flows. In contrast, *Klokov et al.* [22] updated the model with discrete normalizing flows. This leads to faster training and inference.

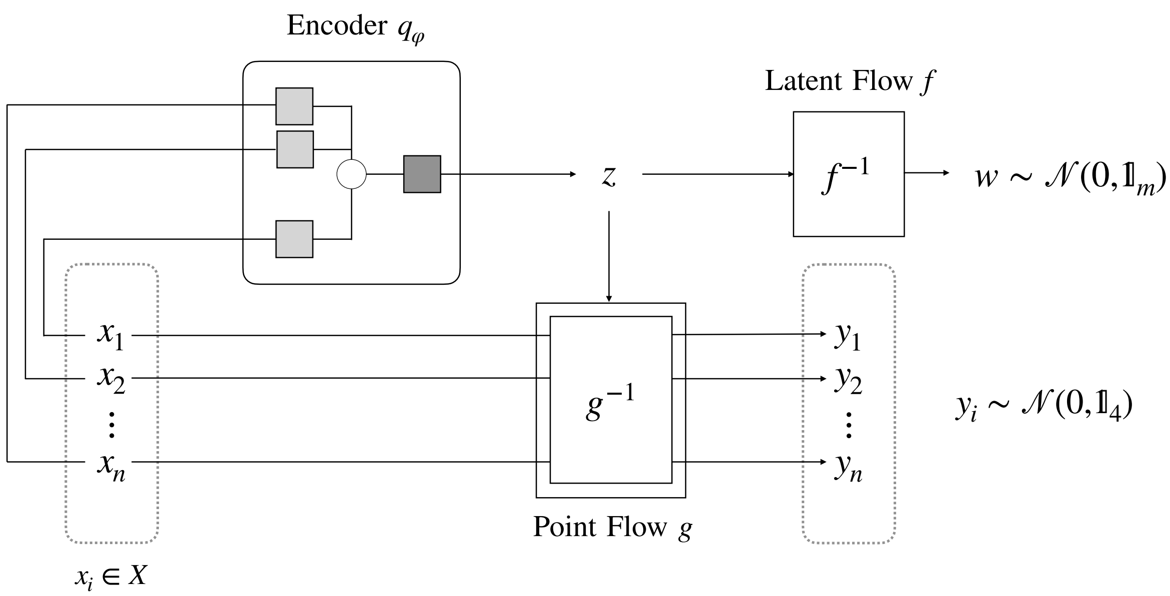

Figure 1. A schematic of the model.

Like both models, our model consists of three sub-models, as shown in Fig. 1. The permutation invariant encoder q_φ, which maps the entire point cloud X into the latent representation z. Our encoder design is based on Particle Flow Networks [23]. The z-representation is enriched with the number of all hits n_{hits} and the transformed total energy of the hits E_{sum}. The other two models are both conditional normalizing flows, based on rational quadratic spline (RQS) coupling layers [24] and ResNet [25] conditioners. To keep the latent space more flexible, z is transformed by the *Latent Flow*, similar to the *Variational Lossy Autoencoder* [26]. The flow is conditioned on the energy of the incoming electron E_{in}. The *Point Flow* transforms each point x_i separately, it is conditioned on the latent variable z, and therefore models the distribution of the points x_i conditioned on the distribution of z.

The model is trained by minimizing the function

$$\begin{aligned}
\mathcal{L} = & \underbrace{-E_{q_\varphi(z|X)}\left[\sum_i^{n_{\text{hits}}} \ln \mathcal{N}(g^{-1}(x_i, z); 0, 1) + \ln \left|\det \frac{\mathrm{d}g^{-1}(x_i, z)}{\mathrm{d}z}\right|\right]}_{\mathcal{L}_{\text{recon}}} \\
& \underbrace{-E_{q_\varphi(z|X)}\left[\ln \mathcal{N}(f^{-1}(z, E_{\text{in}}); 0, 1) + \ln \left|\det \frac{\mathrm{d}f^{-1}(z, E_{\text{in}})}{\mathrm{d}z}\right|\right]}_{\mathcal{L}_{\text{prior}}} \\
& + \underbrace{\left(-\frac{d}{2}(1 + \ln(2\pi)) - \sum_{i=1}^{d} \ln \sigma_i\right)}_{\mathcal{L}_{\text{entr}}}
\end{aligned} \tag{1}$$

with the ADAM [27] optimizer. The loss function at hand is based on the loss function of a variational autoencoder (ELBO). Since normalizing flows are invertible, $\mathcal{L}_{\text{recon}}$ is not trained on reconstructing the point clouds. Instead, the model is trained to maximize the likelihood of the transformed data to a Gaussian. With $\mathcal{L}_{\text{prior}}$ the model learns a transformation from a Gaussian to the latent space z. The last part is the entropy of the decoder and acts as a regularization term on the z space.

All shower hits are generated in parallel and independently of each other. The model cannot take into account whether a calorimeter cell has already been taken by another generated shower hit. Therefore, we need a special sampling method to get consistent showers.

For shower generation, we want to sample from the probability density of the possible showers. Therefore, we need the probability that a calorimeter cell was hit and the probability density of the possible energies in the hit cell. Both quantities are not directly accessible. We approximate these probabilities with the Point Flow. We sample a relatively large number of points (10k) with their corresponding probabilities. The average probability of all points in one cell is taken as the probability of hitting that cell. We sample n_{hits} cells without replacement and with their probability. For each sampled cell, we then pick an energy value according to all energy values sampled for the hit cell and their probability. Therefore, we use the model as a Monte Carlo approximator of the showers.

3. Experimental Setup

The CLIC calorimeter data set [10, 28] is used to test the model. In the data, the showering of electrons entering a section of the CLIC detector were simulated in GEANT4 [29]. The energy of the incoming electrons is between 10 – 510 GeV. The ECAL part of the simulation is used. The resulting calorimeter images have a dimensionality of $(51 \times 51 \times 25)$.

The voxel dataset is transformed into a point cloud dataset. All voxels with energy input are considered. The 3 position coordinates are uniformly distributed over the voxel space. So that the resulting distribution fills a unit box. This processing is referred to as *dequantization* [30]. Since the energy inputs are greater than zero and decrease exponentially, the logarithm of the energy is used. The resulting 4D point clouds were then normalized. The model is trained using four parallel NVIDIA V100 GPUs for a total time of 24 hours.

4. Results and Discussion

For generative models to replace simulations, their outputs must be consistent. To validate this, we look at different distributions of the generated data.

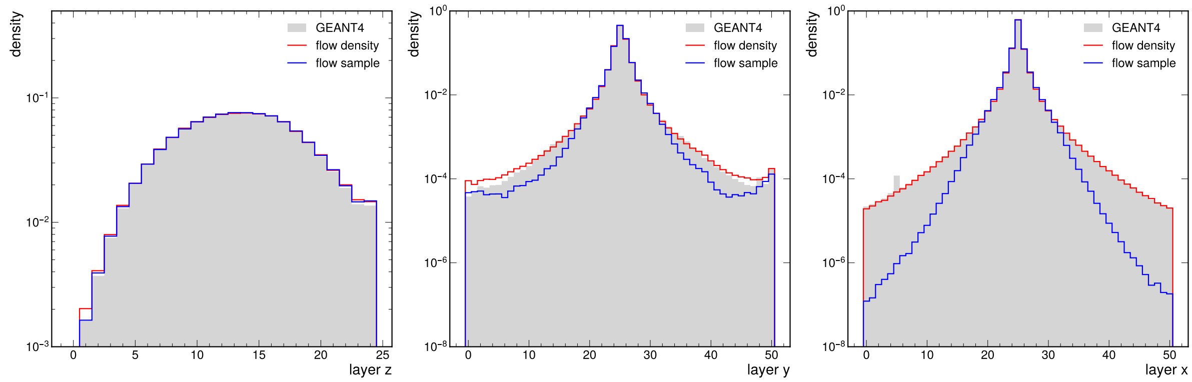

Figure 2. the average shower profiles in all three directions are shown. In gray, the one of GEANT4. In red, the direct density of the flow. In blue, the result of the sampling.

Fig. 2 shows the average shower profile in all directions. The results of the direct density of the model and the results after sampling from the model are compared with the simulation data. It can be seen that the model produces matching results, but the tails of the distributions are not well represented by the sampling.

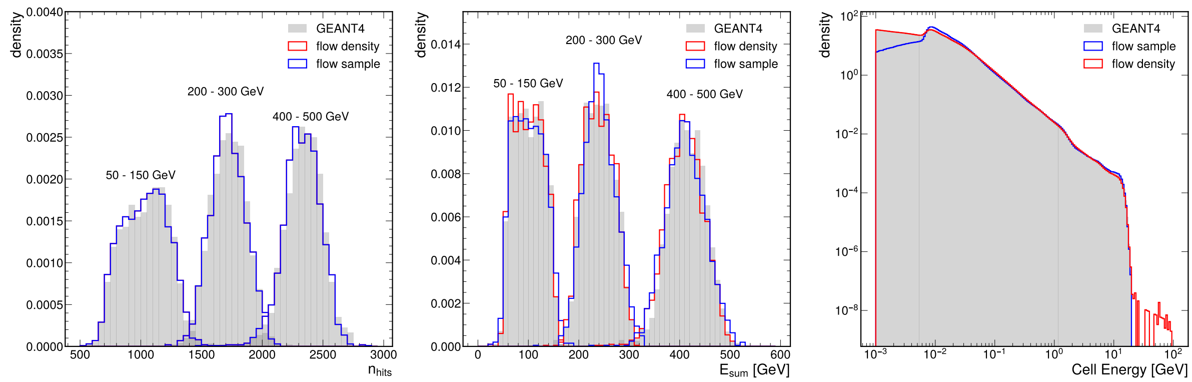

Figure 3. Three statistics of the data are shown. On the left, the number of cells with energy in three energy regimes. In the middle, the summed energy in the cells in the same regimes. On the right, the distribution of energy inputs in the cells. The colour scheme is consistent with Fig. 2.

Fig. 3 compares three statistics of the showers. The first two show that both the number of cells hit and the sum of the energy in the cells agree well with the results of GEANT4 – both for the direct density of the model and for the sampled results. The right graph shows that there are significantly fewer low-energy hits after sampling. This is consistent with the decrease in the tails in Fig. 2.

5. Conclusion and Outlook

The results of the model appear promising. Except for the tails, the model generates showers of a high quality. A possible further development to get the problems of the model at the tails under control would be the use of a post-processing network, as shown in [13, 14]. We are currently investigating the model's performance on other datasets. This will be part of another publication.

Overall, the model shows good results and can overcome the problems of voxel based models.

Acknowledgement

This research was funded by Deutsches Elektronen-Synchrotron DESY, a member of the Helmholtz Association (HGF). This research was supported through the Maxwell computational resources operated at DESY.

References

[1] Albrecht et al. A roadmap for hep software and computing r&d for the 2020s. 2019.

[2] Aberle et al. *High-Luminosity Large Hadron Collider (HL-LHC): Technical design report.* 2020.

[3] de Oliveira et al. Learning particle physics by example: Location-aware generative adversarial networks for physics synthesis. 2017.

[4] Paganini et al. Accelerating science with generative adversarial networks: An application to 3d particle showers in multilayer calorimeters. 2018.

[5] Paganini et al. CaloGAN: Simulating 3d high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks. 2018.

[6] Vallecorsa. Generative models for fast simulation. 2018.

[7] Chekalina et al. Generative models for fast calorimeter simulation: the LHCb case&gt. 2019.

[8] Erdmann et al. Precise simulation of electromagnetic calorimeter showers using a wasserstein generative adversarial network. 2019.

[9] Khattak et al. Fast simulation of a high granularity calorimeter by generative adversarial networks, 2021.

[10] Belayneh et al. Calorimetry with deep learning: particle simulation and reconstruction for collider physics. 80(7), 2020.

[11] Rehm et al. Reduced precision strategies for deep learning: A high energy physics generative adversarial network use case. 2021.

[12] Goodfellow et al. Generative adversarial networks, 2014.

[13] Buhmann et al. Getting high: High fidelity simulation of high granularity calorimeters with high speed. 2021.

[14] Buhmann et al. Hadrons, better, faster, stronger, 2021.

[15] Rezende et al. Variational inference with normalizing flows, 2015.

[16] Krause et al. Caloflow: Fast and accurate generation of calorimeter showers with normalizing flows, 2021.

[17] Krause et al. Caloflow ii: Even faster and still accurate generation of calorimeter showers with normalizing flows, 2021.

[18] The Phase-2 Upgrade of the CMS Endcap Calorimeter. Technical report, 2017.

[19] CALICE Collaboration. Hadronic energy resolution of a combined high granularity scintillator calorimeter system, 2018.

[20] Mikuni et al. Score-based generative models for calorimeter shower simulation, 2022.

[21] Yang et al. Pointflow: 3d point cloud generation with continuous normalizing flows, 2019.

[22] Klokov et al. Discrete point flow networks for efficient point cloud generation, 2020.

[23] 2019.

[24] Durkan et al. Neural spline flows, 2019.

[25] He et al. Deep residual learning for image recognition, 2015.

[26] Chen et al. Variational lossy autoencoder, 2016.

[27] Kingma et al. Adam: A method for stochastic optimization, 2014.

[28] Pierini et al. CLIC Calorimeter 3D images: Electron showers at Fixed Angle, January 2020.

[29] Agostinelli et al. GEANT4–a simulation toolkit. *Nucl. Instrum. Meth. A*, 506:250–303, 2003.

[30] Papamakarios et al. Normalizing flows for probabilistic modeling and inference. 2019.