CaloLatent: Score-based Generative Modelling in the Latent Space for Calorimeter Shower Generation

Thandikire Madula
Department of Physics and Astronomy
University College London
London WC1E 6BT, United Kingdom
thandi.madula.17@ucl.ac.uk

Vinicius M. Mikuni
National Energy Research Scientific Computing Center
Berkeley Lab, Berkeley, CA 94720, USA
vmikuni@lbl.gov

Abstract

Fast calorimeter simulation is crucial for collider physics to accelerate the comparisons between theory and experiments. Deploying physics-based simulators to model high granularity detector responses is slow. Fast surrogate models based on machine learning have shown great promise; they are able to leverage modern computational hardware and can capture the complex and high dimensional space of calorimeter detectors. In this paper we introduce a new fast surrogate model based on latent diffusion models named CaloLatent, able to reproduce, with high fidelity, the detector response in a fraction of the time required by similar generative models. We evaluate the generation quality and speed using the Calorimeter Simulation Challenge 2022 dataset.

1 Introduction

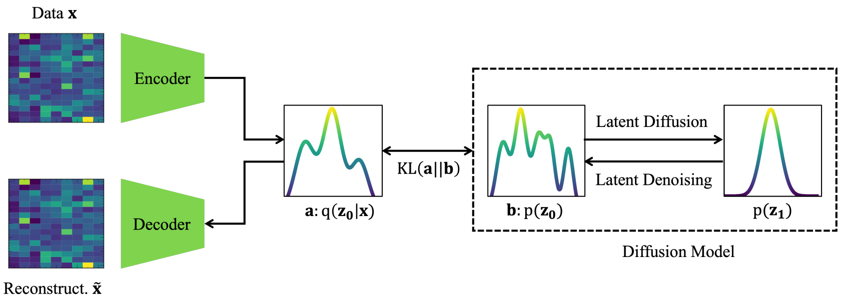
Data $\mathbf{x}$
Encoder
Decoder
Reconstruct. $\tilde{\mathbf{x}}$
$\mathbf{a}$: $q(\mathbf{z_0}|\mathbf{x})$
$\mathrm{KL}(\mathbf{a}||\mathbf{b})$
$\mathbf{b}$: $p(\mathbf{z_0})$
Latent Diffusion
Latent Denoising
$p(\mathbf{z_1})$
Diffusion Model

Figure 1: A schematic diagram of latent diffusion

Detector simulation is necessary for translating physics predictions to measured observables. Among the different detector components, calorimeter simulation is the most computationally expensive. Standard physics simulators yield accurate shower simulations but are slow due to the high granular detectors and particle interactions causing cascades of secondary particles. In standard physics simulation packages such as GEANT4 [1, 2], the computational complexity becomes prohibitive as

Machine Learning and the Physical Sciences Workshop, NeurIPS 2023.

many millions of simulations are required to match the amount of data collected by experimental collaborations. A diverse number of fast surrogate models for calorimeter simulation using machine learning models have been proposed using Generative Adversarial Networks (GANs) [3–18], Variational Autoencoders (VAEs) [19] [17, 16, 20–22], Normalizing Flows (NFs) [23] [24–30], and Diffusion Models [31] [32–34]. Leveraging hardware accelerators, such as Graphics Processing Units (GPUs), these surrogate models are often hundreds to thousands of times faster than full simulation routines.

This work presents a fast surrogate model using a latent diffusion generative model named CaloLatent. Diffusion models have proven to be capable of high fidelity generation. However, sampling from diffusion models is known to be slower than other generative models, e.g. GANs. Conversely, VAEs are quick to sample from but often produce lower quality samples due to remaining non-gaussianities in the latent space. In latent diffusion we utilise a VAE to obtain a lower dimensional latent representation of the data. Once the VAE has learnt an effective latent representation, we can deploy the diffusion model to learn the distribution of the latent space.

Figure 1 provides an overall schematic of how these two generative models are interlaced to compose the latent diffusion model. This configuration affords us two advantages. First it allows us to train the VAE with a smaller KL regularisation term, meaning the VAE will focus more on accurately reconstructing the data. The non-gaussianity of the learned latent space is not prohibitive to the generation abilities of the model, as the diffusion model should allow us to still sample from a gaussian. Secondly, as the diffusion model is deployed in the lower dimensional space, the sampling speed should be faster compared to sampling a model trained on the full dimensionality of the data.

2 Method

The training dataset used in this work originates from the 2022 Calorimeter Challenge. We focus on dataset 2 [35], which consists of GEANT4 simulated electron showers, with incident electron energies log-uniformly distributed between 1 GeV and 1 TeV. The detector design in this dataset is a cylindrical detector with 45 layers in the z direction. Each layer consists of 144 cells, 16 in the angular direction and 9 in the radial direction. The preprocessing applied to the dataset follows that of CaloScore 2 [32].

CaloLatent is composed of three neural networks. First, we have a score-based diffusion model that exclusively learns the energy deposition per layer. A ResNet architecture with 3 layers consisting of 512 hidden nodes forms the foundation of the layer model. The model is conditioned on the incident particle energy and time step information. Next is the VAE. The VAE architecture utilises 3D convolutional neural network (CNN) with a ResNet-like structure. Two down sampling blocks, consisting of a ResNet block followed by an average pooling layer, are used in the VAE encoder. The number of nodes in the down sampling blocks are 64 and 128, an attention mechanism is used in the first block. After the down sampling blocks there are two ResNet blocks with 256 hidden nodes. The decoder of the VAE is a mirror of the encoder but deploying upsampling in place of pooling. Conditional inputs, incident particle energy and energy per layer, are used as additional inputs to the VAE. The final model is the score-based diffusion model used to learn the distribution of the of the latent space. The architecture of this model is identical to the layer model; however, it is additionally conditioned on the energy per layer.

The three models comprising CaloLatent are trained independently using tensorflow [36] and keras [37], the layer model and the VAE are trained in parallel for 500 epochs. Subsequently, the latent diffusion is trained for 350 epochs. We train two VAEs, the first with the KL divergence weighted by a factor of $1\mathrm{e}{-6}$ and the second with the full KL divergence. We use the VAE trained with the full KL divergence as a baseline to evaluate the performance gain provided by deploying the diffusion model in the latent space. The models are trained using 4 A100 Nvidia GPUs. All models are trained with a cosine learning rate schedule with an initial learning rate of $4\mathrm{e}{-4}$.

3 Results

To evaluate the performance of the latent diffusion model we use a combination of metrics provided by the Calorimeter Simulation Challenge 2022 and an additional classifier test to help qualify the improvement using CaloLatent has over training a vanilla VAE. For both the vanilla VAE and

CaloLatent we generate 100K showers. Figure 2 presents some key distributions used to evaluate how well the distributions of the generated showers match those of GEANT4 . The top row of Figure 2 shows the mean energy deposition in r, α and z. The distributions in r and α are generated by the VAE and CaloLatent, the distribution in z is generated by the independent energy per layer model. Overall, we can see that both the vanilla VAE and CaloLatent have the best agreement with GEANT4 in the α direction and both models struggle the most in the r direction. In the directions predicted by the VAE and CaloLatent , r and α, we can see that the VAE does slightly better than CaloLatent with a combined earth movers distances of 0.87 and 0.1 respectively. The lower the EMD between two distributions the closer they are to each other.

The second row of Figure 2 presents the angular distributions of the calorimeter shower in terms of angular widths. The shower width in a given coordinate, r or α, can be calculated using equation 1.

$$\sigma = \sqrt{\langle x^2 \rangle - \langle x \rangle^2}, \qquad \langle x \rangle = \frac{\sum_j x_j E_j}{\sum_j E_j} \tag{1}$$

Where σ is the shower width and $\langle x \rangle$ is the energy weighted mean.

As can be seen in the bottom left plot of Figure 2 both the vanilla VAE and CaloLatent are able to model the shower width in r well with CaloLatent staying within 10% of GEANT4 across the distribution range and the vanilla VAE only being outside the 10% band at the tails. Furthermore, for this shower width, CaloLatent provides visibly better performance, quantified by an EMD value 0.01 vs 0.03. The bottom middle plot of Figure 2 presents the shower widths in α, the modelling of this variable is similar between models when evaluated using the EMD, both having a value of 0.03, however it can be seen that the shape of the distribution generated by CaloLatent is closer to the GEANT4 distribution demonstrated by the flatness of the line in the % difference panel. Finally, in the bottom right plot of the figure the distribution of the total deposited energy is presented. Both the vanilla VAE and CaloLatent distributions agree well with GEANT4 .

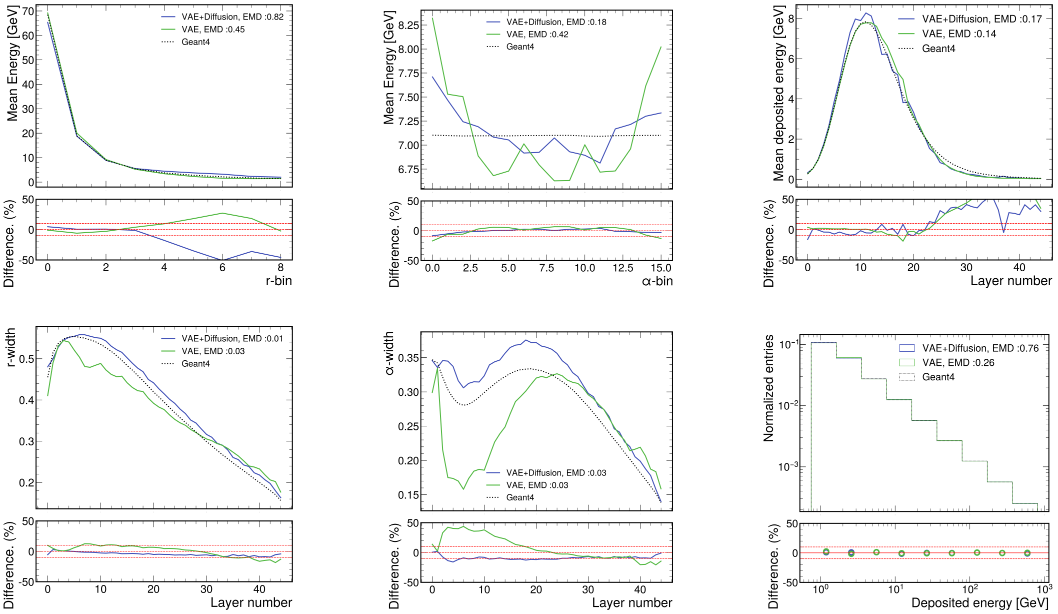

Figure 2: Mean deposited energies in the r- (top left), α- (top middle), and z- directions (top right). Particle shower widths in the r- (bottom left) and α- directions (bottom middle). The sum of all voxel energies (bottom right).

An alternative approach for evaluating generative models in high dimensions is to utilize classifiers. To evaluate the performance of CaloLatent, two types of classifiers are trained and evaluated, type I aims to discriminate between showers generated by the vanilla VAE, class label 0, from CaloLatent showers, class label 1. Type II aims to distinguish showers generated from the generative models

Table 1: Area under the ROC curve (AUC) and Jensen- Shannon divergence (JSD) for classifiers trainined using low level and high level information

	AUC/JSD	
Classifier Inputs	VAE	VAE + Diffusion
High level	0.9951 / 0.8748	0.9865 / 0.7868
Low level	0.9947 / 0.8907	0.9808 / 0.7614
Normalised low level	0.9924 / 0.8462	0.9662 / 0.6595

from GEANT4 samples. We trained the type I classifier on the low-level shower voxels. The model was a simple feedforward neural network with 2 hidden layers, each containing 256 nodes. The area under the roc curve (AUC) for the model was 0.9979. Figure 3 shows the output probabilities of the classifier when evaluated on VAE samples, CaloLatent samples and GEANT4 samples. The average classifier output for the GEANT4 samples was 0.9227, this means, to this classifier, GEANT4 samples are closer to CaloLatent samples than VAE samples.

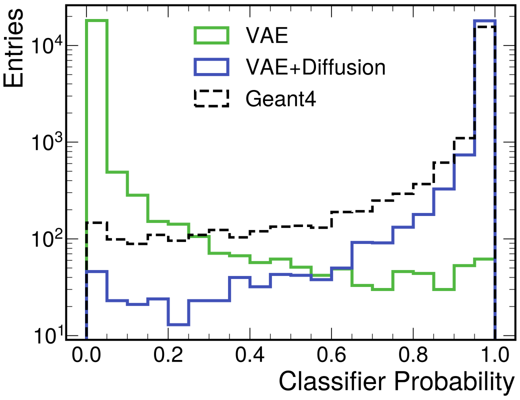

Figure 3: Classifier output probabilities for binary classifier trained to distinguish between VAE and CaloLatent Samples.

A limitation of the type I classifier is that it is strictly comparative, it makes no absolute statement about how good either model is at generating samples that look close to the truth. To complement the type I classifier we train type II classifiers, to this end we utilize the official classifier tests provided by the challenge and results are summarised in Table 1. A perfect generative model would have an AUC of 0.5 and a Jesen-Shanonn divergence (JSD) of 0. Using both high level - histogram - and low level - voxel - inputs, we can see from Table 1 that both the VAE and CaloLatent have AUC scores close to 1. This means that a classifier is able to distinguish them from GEANT4 samples. They also have relatively high JSD values, however the AUC and JSD values are lower for CaloLatent than the VAE regardless of the type of input used. This suggests, in a different perspective that CaloLatent produces better samples. Although the classfier AUCs are quite high, it is important to note that classifiers maybe sensitive to shower differences that are not physically important.

A critical aim of this work was to develop a surrogate model that is faster than similar published models. CaloLatent is able to generate 100 showers in 1.9 s compared to CaloScore which takes 5.8 s and CaloScore 2 [32] which takes 27.8 s. It is important to note, CaloScore 2 [32] deploys a distillation technique which is able to reduce its generation time to 0.010 s. However, this requires the training of an additional teacher-student model.

4 Conclusion

This work presents CaloLatent, a latent diffusion based fast surrogate model that is able to improve on the generation quality of a traditional VAE and improve on the sampling speeds of score-based diffusion models. In latent diffusion models, often the performance bottle neck arises from the quality of the latent space produced by the VAE, therefore the main avenue we aim to explore in improving the overall performance of CaloLatent is further developing the VAE backbone.

References

[1] Allison, J. *et al.* Geant4 developments and applications. *IEEE Transactions on Nuclear Science* **53**, 270–278 (2006).

[2] Allison, J. *et al.* Recent developments in geant4. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **835**, 186–225 (2016). URL `https://www.sciencedirect.com/science/article/pii/S0168900216306957`.

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

[4] Paganini, M., de Oliveira, L. & Nachman, B. CaloGAN : Simulating 3D high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks. *Phys. Rev. D* **97**, 014021 (2018). `1712.10321`.

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

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

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

[8] Belayneh, D. *et al.* Calorimetry with Deep Learning: Particle Simulation and Reconstruction for Collider Physics (2019). `1912.06794`.

[9] Vallecorsa, S., Carminati, F. & Khattak, G. 3D convolutional GAN for fast simulation. *Proceedings, 23rd International Conference on Computing in High Energy and Nuclear Physics (CHEP 2018): Sofia, Bulgaria, July 9-13, 2018* **214**, 02010 (2019).

[10] Ahdida, C. *et al.* Fast simulation of muons produced at the SHiP experiment using Generative Adversarial Networks (2019). `1909.04451`.

[11] Chekalina, V. *et al.* Generative Models for Fast Calorimeter Simulation.LHCb case. *CHEP 2018* (2018). `1812.01319`.

[12] Carminati, F. *et al.* Three dimensional Generative Adversarial Networks for fast simulation. *Proceedings, 18th International Workshop on Advanced Computing and Analysis Techniques in Physics Research (ACAT 2017): Seattle, WA, USA, August 21-25, 2017* **1085**, 032016 (2018).

[13] Vallecorsa, S. Generative models for fast simulation. *Proceedings, 18th International Workshop on Advanced Computing and Analysis Techniques in Physics Research (ACAT 2017): Seattle, WA, USA, August 21-25, 2017* **1085**, 022005 (2018).

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

[15] Deja, K., Trzcinski, T. & Graczykowski, u. Generative models for fast cluster simulations in the TPC for the ALICE experiment. *Proceedings, 23rd International Conference on Computing in High Energy and Nuclear Physics (CHEP 2018): Sofia, Bulgaria, July 9-13, 2018* **214**, 06003 (2019).

[16] Deep generative models for fast photon shower simulation in ATLAS (2022). `2210.06204`.

[17] Deep generative models for fast shower simulation in ATLAS. *ATL-SOFT-PUB-2018-001* (2018). URL `http://cds.cern.ch/record/2630433`.

[18] Aad, G. *et al.* AtlFast3: the next generation of fast simulation in ATLAS. *Comput. Softw. Big Sci.* **6**, 7 (2022). `2109.02551`.

[19] Kingma, D. P. & Welling, M. Auto-Encoding Variational Bayes. *arXiv e-prints* arXiv:1312.6114 (2013). `1312.6114`.

[20] Buhmann, E. *et al.* Decoding Photons: Physics in the Latent Space of a BIB-AE Generative Network (2021). `2102.12491`.

[21] Buhmann, E. *et al.* Hadrons, Better, Faster, Stronger (2021). `2112.09709`.

[22] Diefenbacher, S. *et al.* New Angles on Fast Calorimeter Shower Simulation (2023). `2303.18150`.

[23] Rezende, D. & Mohamed, S. Variational inference with normalizing flows. In Bach, F. & Blei, D. (eds.) *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37 of *Proceedings of Machine Learning Research*, 1530–1538 (PMLR, Lille, France, 2015). URL `https://proceedings.mlr.press/v37/rezende15.html`.

[24] Krause, C. & Shih, D. CaloFlow: Fast and Accurate Generation of Calorimeter Showers with Normalizing Flows (2021). `2106.05285`.

[25] Krause, C. & Shih, D. CaloFlow II: Even Faster and Still Accurate Generation of Calorimeter Showers with Normalizing Flows (2021). `2110.11377`.

[26] Buckley, M. R., Krause, C., Pang, I. & Shih, D. Inductive CaloFlow (2023). `2305.11934`.

[27] Krause, C., Pang, I. & Shih, D. CaloFlow for CaloChallenge Dataset 1 (2022). `2210.14245`.

[28] Diefenbacher, S. *et al.* L2LFlows: Generating High-Fidelity 3D Calorimeter Images (2023). `2302.11594`.

[29] Cresswell, J. C. *et al.* CaloMan: Fast generation of calorimeter showers with density estimation on learned manifolds. In *36th Conference on Neural Information Processing Systems* (2022). `2211.15380`.

[30] Liu, J., Ghosh, A., Smith, D., Baldi, P. & Whiteson, D. Generalizing to new geometries with Geometry-Aware Autoregressive Models (GAAMs) for fast calorimeter simulation (2023). `2305.11531`.

[31] Song, Y. *et al.* Score-Based Generative Modeling through Stochastic Differential Equations. *arXiv e-prints* arXiv:2011.13456 (2020). `2011.13456`.

[32] Mikuni, V. & Nachman, B. Score-based generative models for calorimeter shower simulation. *Phys. Rev. D* **106**, 092009 (2022). `2206.11898`.

[33] Buhmann, E. *et al.* CaloClouds: Fast Geometry-Independent Highly-Granular Calorimeter Simulation (2023). `2305.04847`.

[34] Acosta, F. T. *et al.* Comparison of Point Cloud and Image-based Models for Calorimeter Fast Simulation (2023). `2307.04780`.

[35] Giannelli, F. *et al.* Fast calorimeter simulation challenge 2022 - dataset 2 (2022). URL `https://doi.org/10.5281/zenodo.6366271`.

[36] Abadi, M. *et al.* Tensorflow: A system for large-scale machine learning. In *OSDI*, vol. 16, 265–283 (2016).

[37] Chollet, F. Keras. `https://github.com/fchollet/keras` (2017).