Evaluating Generative AI Inference Performance on AI for Science Calorimeter Simulations

Farzana Yasmin Ahmad, Vanamala Venkataswamy, Geoffrey Fox
Biocomplexity Institute, University of Virginia, Charlottesville, VA 22903, USA.
fa7sa@virginia.edu, vv3xu@virginia.edu, gcfexchange@gmail.com

1 Introduction

Replacing traditional HPC computations with deep-learning surrogate models can dramatically improve the performance of simulations. Fast simulation of 3D high-energy particle showers in a multi-layer calorimeter [20] with surrogate models [2] has led to a proliferation of scientific articles in recent years. The goal is to spur the development of fast and high-fidelity calorimeter shower generation using deep-learning methods. Currently, generating calorimeter showers of interacting particles (mainly electrons, photons, and pions) using Geant4 [1] is a major computational bottleneck at LHC [5], and it is expected to overwhelm the computing budget of LHC experiments shortly with the introduction of high-granularity calorimeters. Therefore there is an urgent need to develop Geant4 emulators that are both fast and accurate.

While many recent deep-learning-based surrogate model implementations claim to perform at par with Geant4, our personal experience is that most of these implementations are not open-source. Many open-source implementations do not work (fail to run the code or errors due to missing packages/code). Of the open-source implementations, we were able to reproduce the results for four implementations, namely CaloINN [6], CaloFlow [11], CaloDiffusion [3], and CaloScore [14] [15].

In addition to running these open-source implementations, we plan to share trained models with other collaborators and scientists, enabling the community to independently benchmark and perform a clean and fair comparison of various surrogate models.

2 Calorimeter Surrogates and Datasets

Figure 1 shows the taxonomy of current Calorimeter Shower Simulation methodologies. The development of accurate models for simulating calorimeter showers is crucial for the advancement of particle physics research. However, comparing the performance of existing models is a challenging task due to the diverse range of approaches and metrics. To address this, there is a need for a uniform means of evaluating these models, both qualitatively and quantitatively. Without such an approach, it becomes difficult to build upon existing work or identify areas that require additional effort. Therefore, it is imperative that we work towards developing a standardized approach to evaluate the efficacy of these models to ensure that we can continue to make progress in this field.

2.1 Background

In this section, we briefly describe the generative models evaluated for this study.

2.1.1 Normalizing Flow Based Models

To address the challenges encountered by GAN-based models [18, 7] including instability in training and mode collapse, Normalizing Flow presents a promising alternative for both density estimation and sampling.

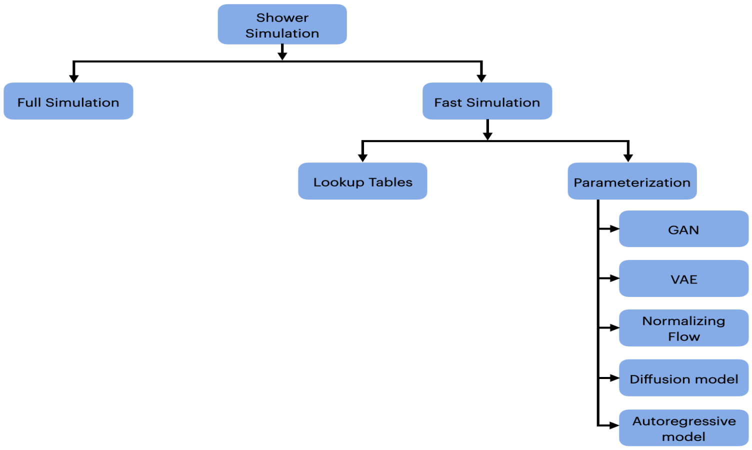


Figure 1: Taxonomy of the calorimeter shower simulation methodologies.

Normalizing Flow learns the bijective transformation between the data space x and the base or latent space z. The data space x is unknown and we want to determine the density of data space x defined as $p(x)$. Normalizing Flow was first introduced in [10, 11, 12] based calorimeter shower simulation called CaloFlow. CaloFlow performs exceptionally well in generating accurate shower samples for the CaloGAN dataset and Dataset-1 of CaloChallenge-2022. Although CaloFlow-I is 50x faster than Geant4, it is 500x slower compared to CaloGAN [18]. This is due to the use of Masked Autoregressive Flow(MAF) [19] which is faster in density estimation but slower in sampling. On the other hand, Inverse Autoregressive Flow(IAF) [9], is faster in sampling but slower in density estimation. CaloFlow-II is faster due to the Probability Density Distillation or teacher-student training, where the IAF (student) is trained on the output of the MAF(teacher). MAF can map the data distribution to latent space and IAF can map the latent space to data distribution faster.

CaloINN [6] is an example of an Invertible Neural Network(INN). Unlike CaloFlow, CaloINN uses coupling flow instead of autoregressive flows. Coupling flows are faster in both density estimation and sampling. As a result, it can overcome the higher dimensionality issue introduced in CaloFlow. In general, CaloINN can not be executed on Dataset 3 of CaloChallenge 2022 due to memory constraints. To achieve this, it combines INN with Variational Autoencoder(VAE). First, it generates a latent space using the VAE and executes INN on this latent space. After sampling from the INN, it is passed through the decoder of the VAE to regenerate the data space. This model fails to demonstrate improvements in the quality of shower generation compared to other state-of-the-art models. Nevertheless, the authors assert that CaloINN is faster when processing high-dimensional data.

2.1.2 Diffusion and Score Based Models

In the Diffusion model, Gaussian noise is added iteratively to the data points. After sufficiently large steps T (the number of diffusion steps), the original data distribution is modified into a multivariate Gaussian distribution. A new sample from the original data distribution can be generated by sampling from the multivariate Gaussian distribution and inverting the diffusion process (to reverse distribution) which is done using approximation. A similar approach is followed by the score-based models, e.g., CaloScore [14] [15]. Although these models show an impressive performance in generating precise shower events, they are slower in sampling compared to other surrogate models. This is due to the number of diffusion steps. If the number of diffusion steps is reduced, the quality of the shower will deteriorate and vice versa. This is a trade-off between sampling speed and accuracy. Table 1 lists the number of parameters in each of the models evaluated in this work. Since we were unable to execute CaloINN on Dataset 3, the count of parameters for this model remains undetermined.

Model Name	Dataset	Num of Params
CaloDiffusion	DS1	$\sim$ 520K
	DS2	$\sim$ 520K
	DS3	$\sim$ 1.2M
CaloScore	DS1	$\sim$ 2.25M
	DS2	$\sim$ 3.76M
	DS3	$\sim$ 3.76M
CaloINN	DS1(photon)	$\sim$ 18.8M
	DS1(pion)	$\sim$ 26.5M
	DS2	$\sim$ 193.5M
	DS3	–
CaloFlow(Teacher)	CaloGAN	$\sim$ 37.8M
CaloFlow(Student)	CaloGAN	$\sim$ 50.9M

Table 1: Number of parameters in CaloDiffusion, CaloScore, CaloINN and CaloFlow models.

2.2 Datasets

The CaloChallenge-2022 [13] offers three datasets, ranging in difficulty from easy to medium to hard. The difficulty is set by the dimensionality of the calorimeter showers (the number of layers and the number of voxels in each layer). As the difficulty increases, the surrogate model training time also increases.

All datasets follow a similar layout. The detector geometry comprises concentric cylinders, and the particles travel along the z-axis. The detector is divided into discrete layers along the z-axis, and each layer has bins along the radial direction. Some layers also have bins at the angle α. The binning XML files store the number of layers and the number of bins in r and α, which are read by helper functions. The coordinates $\Delta\varphi$ and $\Delta\eta$ correspond to the x and y-axis of the cylindrical coordinates. Figure 2 illustrates a 3D view of a geometry with three layers, where each layer has three bins in the radial direction and six bins in the angular direction. The right image shows the front view of the geometry, as seen along the z-axis.

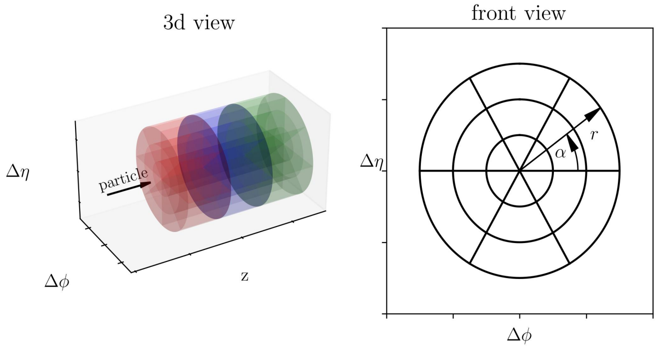


Figure 2: Calorimeter geometry (image source: [13]).

In Dataset-1, the number of radial and angular bins varies from layer to layer and is different for photons and pions, resulting in 368 voxels (in 5 layers) for photons and 533 (in 7 layers) for pions. In Dataset-2, each of the 45 layers has 9 radial bins and 16 angular bins, yielding a total of $45 \times 16 \times 9 = 6480$ voxels. In Dataset-3, each of the 45 layers has 18 radial and 50 angular bins, totaling $45 \times 50 \times 18 = 40500$ voxels. Dataset-2 and Dataset-3 detector geometries are similar with Dataset-3 having a much higher granularity.

CaloGAN dataset [17, 18] is a toy version of Dataset-1 from CaloChallenge-2022. This dataset was introduced in [18] and later used to evaluate [10, 11, 12]. This dataset has three layers and it has Geant4 calorimeter images for photon, pion, and positron. These are all electromagnetic calorimeter showers. Here incident energies vary from $1 - 100GeV$. It has irregular layer resolution where the first, second, and third layer has resolution $3 \times 96, 12 \times 12$, and 12×6 respectively.

3 Evaluation

In this section, we briefly describe the evaluation methodology, a brief description of the computing infrastructure, and evaluation metrics used to compare the performance of various models and the results. We use Geant4 as our baseline to compare the performance of the generative models.

3.1 Evaluation methodology

For CaloDiffusion, we ran the trained model for three different batch sizes mentioned in the table **??**. For each batch size, we executed it three times to take an average shower generation time per event. We could not generate samples for CaloDiffusion on Dataset 3 in CPU due to our resource usage constraint. A similar approach was used to determine the shower generation time per event in CaloINN with the same batch size ranges. We were unable to run CaloINN for dataset 3 even in NVIDIA V100 GPU. The authors of the CaloINN paper faced the same issue and proposed VAE+INN for Dataset 3. We did not show it here. For CaloScore, we ran 10 iterations each iteration sampling 12100 events, which gives the time taken to generate 12100 events. We averaged the 10 iterations and calculated the time taken (in seconds) to generate one event.

All of the GPU inferences were executed on machines with NVIDIA V100 GPUs with 32 GB VRAM. The CPU inferences were executed on Intel(R) Xeon(R) Gold 6248R CPU @ 3.00GHz machines.

The CaloFlow(Teacher) model was executed on the NVIDIA V100 GPU. We considered two batch sizes 10 and 100 respectively. The inference was repeated three times for each batch size and we averaged the generation time per shower. Table 3 lists the sampling time for Geant4 as reported in [18] and CaloFlow(Student) and CaloGAN as reported in [12] where authors used TITAN V GPU to run the inference.

3.2 Evaluation metrics

To evaluate the generative models on the new NVIDIA Blackwell framework (FP4), we propose two metrics; 1) the execution time of the inference and 2) the quality of the inference. The execution time of the inference is the time taken to generate one event, in seconds. The quality of inference specifies how closely the generated showers match that of the showers generated by Geant4. The goal of employing generative models is to minimize the time taken to generate calorimeter showers while maintaining the accuracy of the generated showers as closely as possible to the showers generated by Geant4.

3.2.1 Execution time of the inference

Table 2 lists the inference time (in seconds) per event on CPU and GPU for CaloDiffusion, CaloScore, and CaloINN models. The missing entries in the table indicate that we did not record the time for that configuration either due to hardware limits (some models failed to load on the GPU memory) or we are still waiting for the results. Table 3 lists the inference time (in seconds) per event on CPU and GPU for CaloFlow student and teacher models.

3.2.2 Quality of the inference

One of the most interesting observables to understand the quality of the generated shower is layer energy distribution. Here we are showing layer energy distribution for dataset-2 and dataset-3 as surrogate models usually struggle to capture the distribution when the dimension is higher. Instead of showing layer energy distribution for each 45 layers individually, we grouped five consecutive layers and then computed the layer energy distribution on it shown in figures 3 through 9. To further isolate the performance of the surrogate models in different ranges of incident energy, we separated the graphs into three different energy ranges.

		CaloDiffusion (sec)		CaloScore (sec)		CaloINN (sec)	
Dataset	Batch Size	CPU	GPU	CPU	GPU	CPU	GPU
1(photons)	1	10.61	4.64	–	3.8	–	0.0257
	10	2.79	0.48	–	3.9	–	0.0026
	100	2.99	0.07	–	3.9	–	0.0002
1(pions)	1	11.13	4.77	–	–	0.0306	0.0261
	10	2.96	0.51	–	–	0.0090	0.0027
	100	3.49	0.08	–	–	0.0032	0.0003
2 (electrons)	1	37.21	4.55	–	–	–	0.0428
	10	10.92	0.53	–	–	–	0.0052
	100	22.71	0.22	–	–	–	0.0015
3 (electrons)	1	–	6.02	–	–	–	–
	10	–	2.47	–	–	–	–
	100	–	1.92	–	–	–	–

Table 2: Shower generation time per event for CaloDiffusion on CPU and GPU for various batch sizes

Batch Size	CaloFlow(Teacher)(sec)	CaloFlow(Student)(sec)	CaloGAN(sec)	Geant4 (sec)
10	0.789	0.0058	0.0022	1.772
100	0.141	0.0006	0.0003	1.772

Table 3: Shower generation time per event for CaloFlow(teacher), CaloFlow(student), CaloGAN and Geant4 in CaloGAN dataset

Figures 3 through 5 shows the energy deposited in each calorimeter layer (layers 0 through 44) plotted in groups of 5 for electrons for dataset-2.

Figures 7 through 9 shows the energy deposited in calorimeter each layer (layers 0 through 44) plotted in groups of 5 for electrons for dataset-3.

To quantify the similarities between two different histograms we use separation power. In information theory, it is known as triangular discrimination. To compute separation power, we use the equation 1. In this equation, $h1$ and $h2$ denote two different histograms that we want to compare. Separation power 0 means two histograms are the same and 1 means they do not have any overlapping bins.

$$< S^2 >= \frac{1}{2} \sum_{i=1}^{n_{\text{bins}}} \frac{(h_{1,i} - h_{2,i})^2}{h_{1,i} + h_{2,i}} \tag{1}$$

Figure 11 shows the separation power a.k.a triangular discrimination between Geant4 and the generative models for dataset-2. From the graphs, the separation power between Geant4 and CaloINN is higher than other models meaning CaloINN is less accurate in generating showers. On the other hand, CaloDiffusion and CaloScore both are good at capturing the Geant4 distribution. Among them, CaloScore is performing slightly better than CaloDiffusion. CaloINN performs significantly worse than the other two models.

Figure 12 shows the separation power between Geant4 and the CaloDiffusion for dataset-3.

Figure 13 and 14 show the energy deposited in all three different layers of CaloGAN dataset for positron and photon respectively. We could not compare CaloFlow with other models as this version is specifically implemented for CaloGAN dataset. An improved iteration of CaloFlow, known as Inductive CaloFlow [4], has been developed for implementation on Dataset 2 and 3. We were unable to assess it because it was not accessible online.

4 Conclusion

Fast and accurate simulation of 3D high-energy particle showers in a multi-layer calorimeter with surrogate models is essential leading to substantial reduction in time and computational cost. Fast and high-fidelity calorimeter shower generation using deep-learning methods is making significant progress in this direction. We evaluated four such models namely CaloDiffusion, CaloScore, CaloINN, and CaloFlow. Based on the experimental results, CaloScore and CaloDiffusion generate the most accurate showers with CaloScore being more accurate than CaloDiffusion. In the future, we plan to evaluate CaloQVAE [8], a quantum variational autoencoder based generative model to simulate calorimeter shower. In CaloQVAE, a quantum processing unit(QPU) is used to generate samples. Additionally, we plan to extend this study by evaluating these models on other hardware platforms including the latest NVIDIA Blackwell framework [16]. We hope to explore the Blackwell framework's lower floating point precision (FP4) to study its effects on accuracy and computational time.

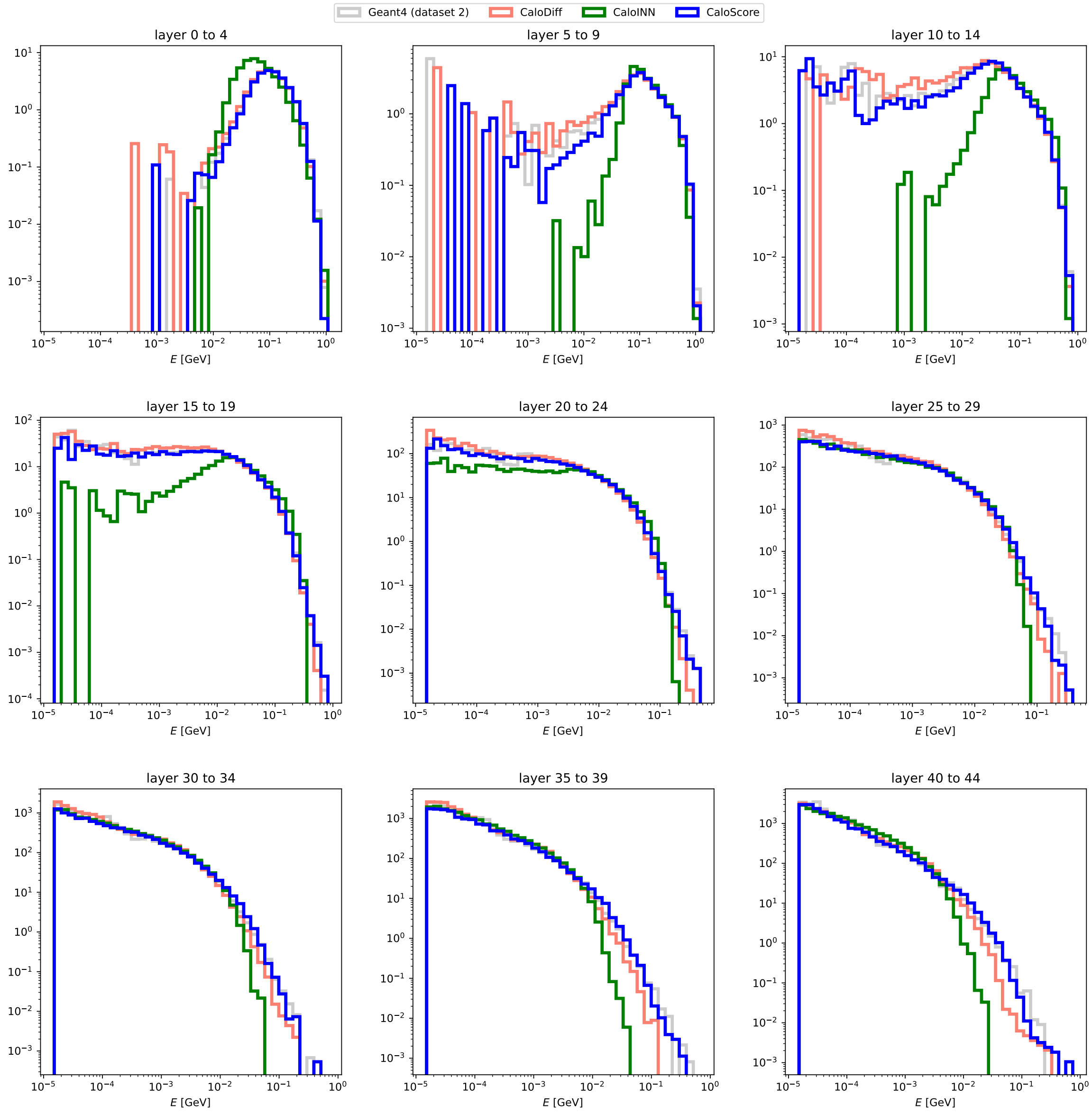


Figure 3: Layer energy (1GeV to 10GeV) for Dataset-2.

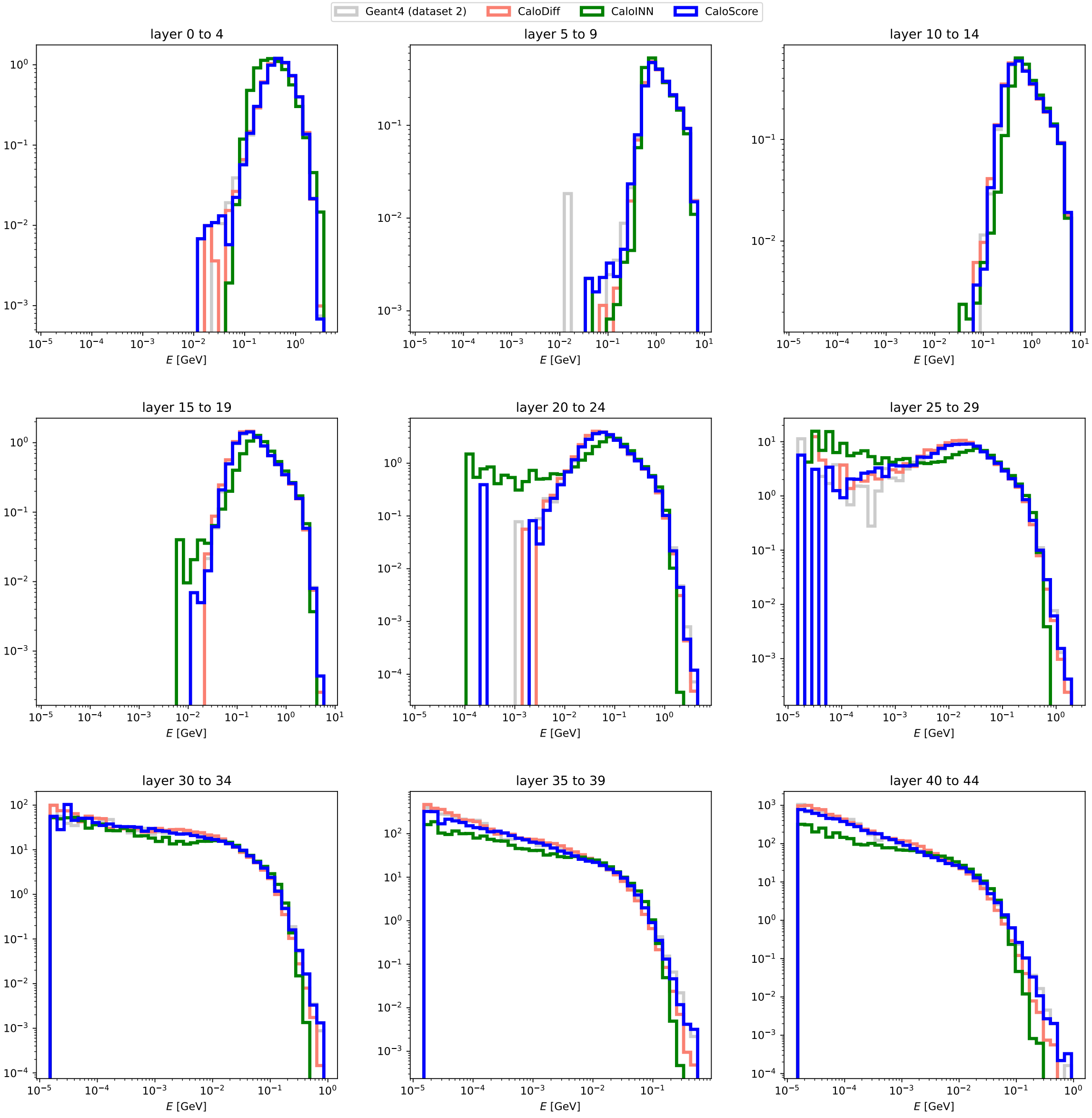


Figure 4: Layer energy (10 GeV to 100 GeV) for Dataset-2.

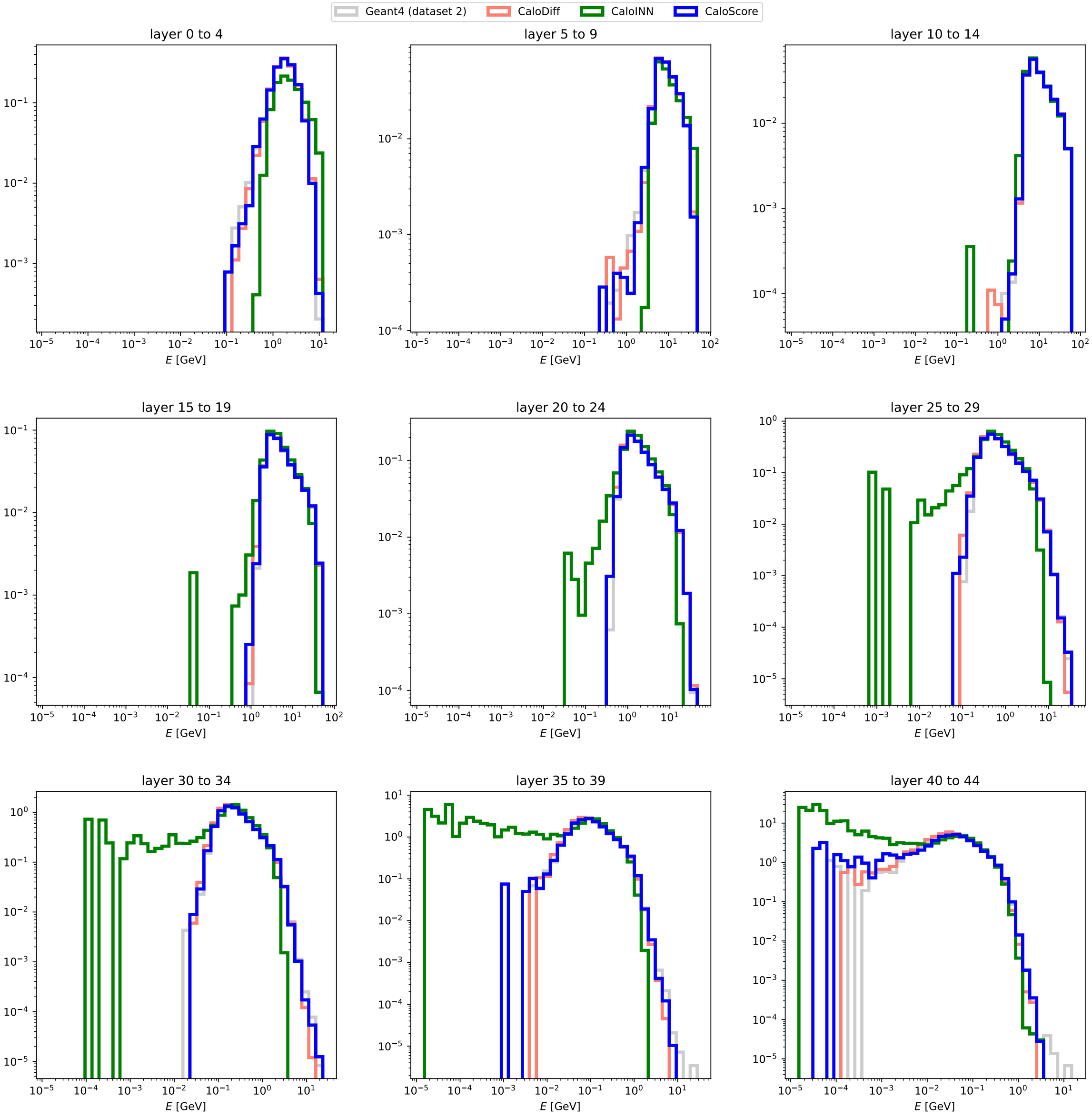


Figure 5: Layer energy (100 GeV to 1000 GeV) for Dataset-2.

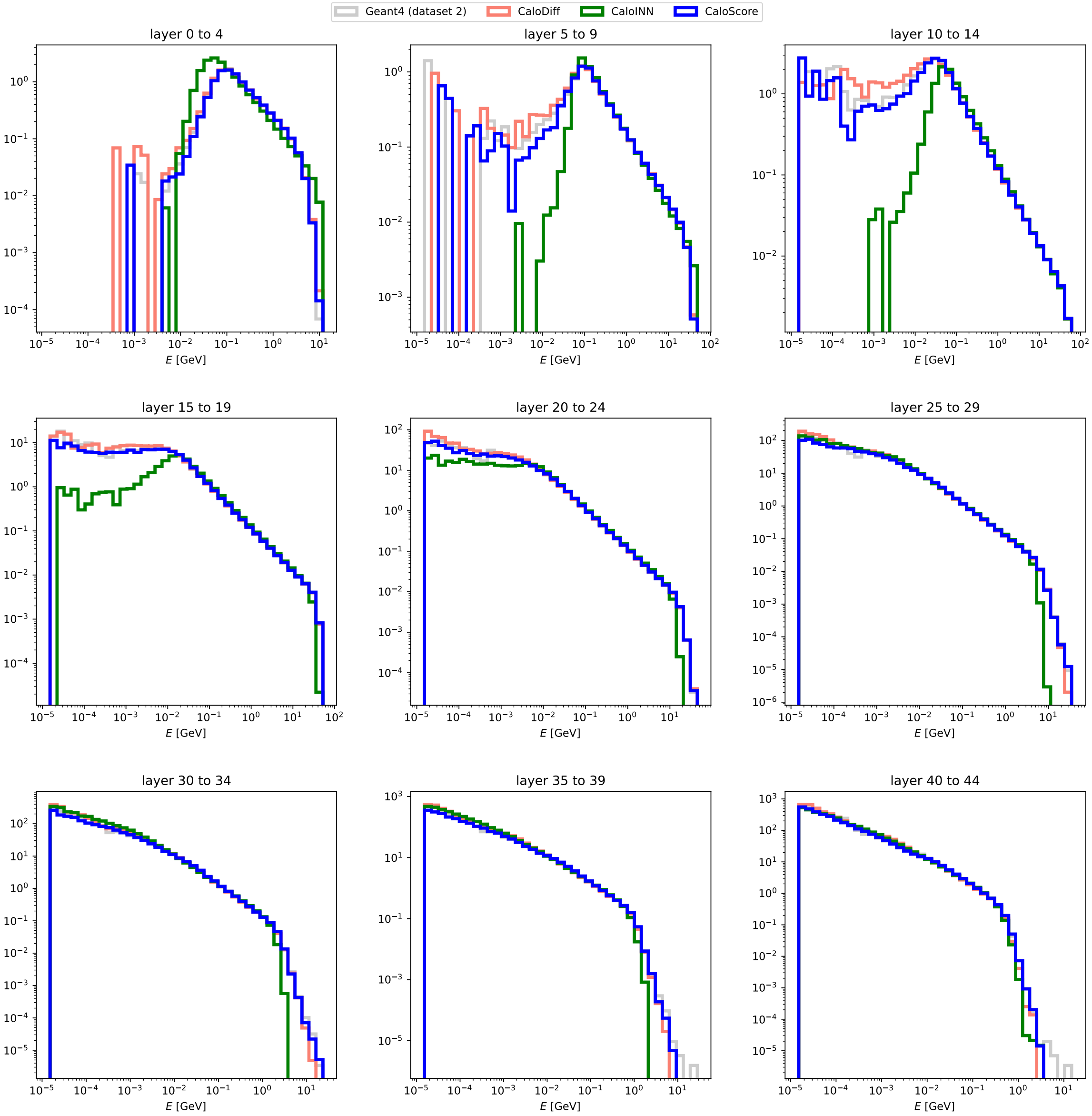


Figure 6: Layer energy (1 GeV to 1000 GeV) for Dataset-2.

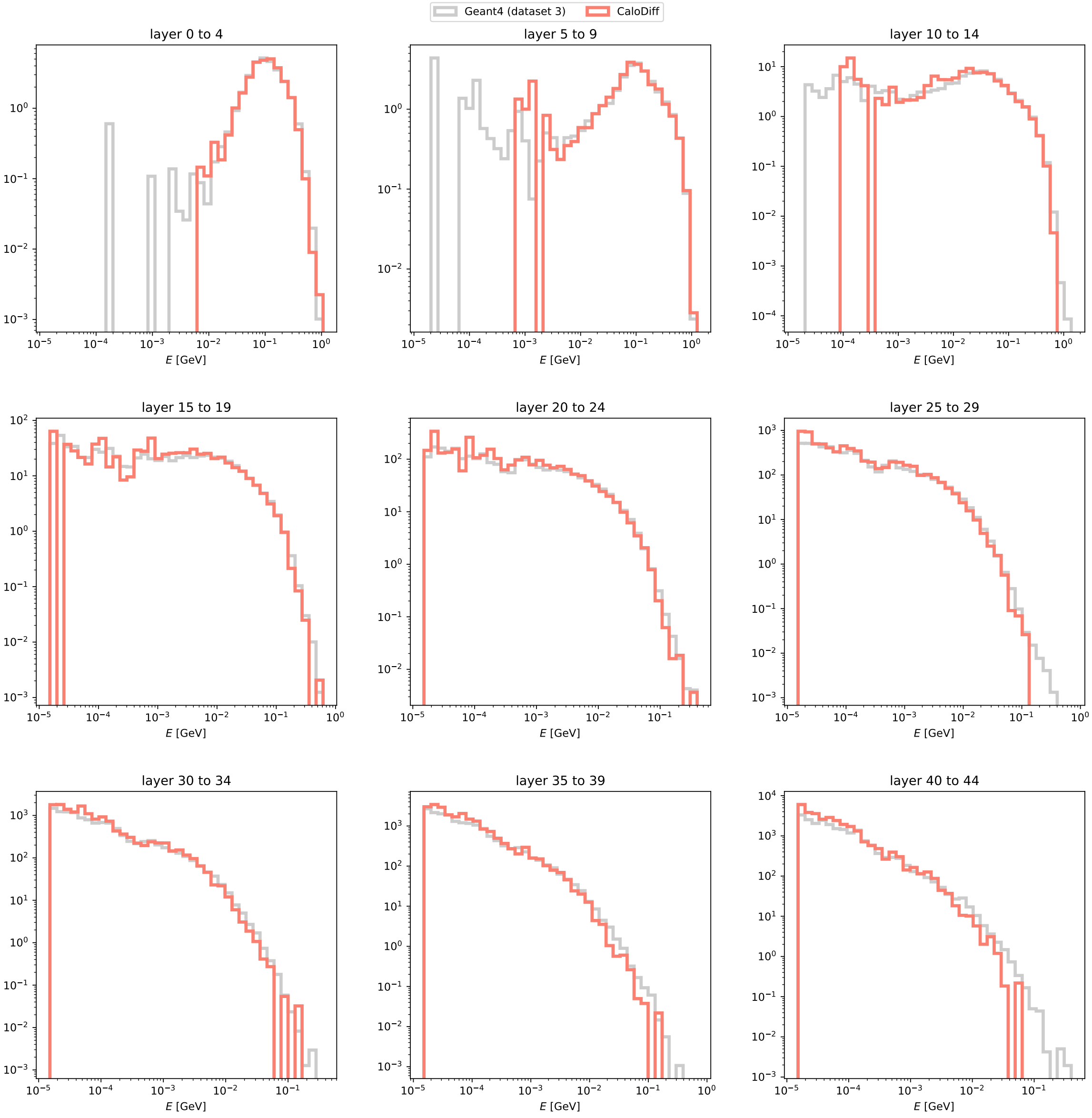


Figure 7: Layer energy (1GeV to 10GeV) for Dataset-3.

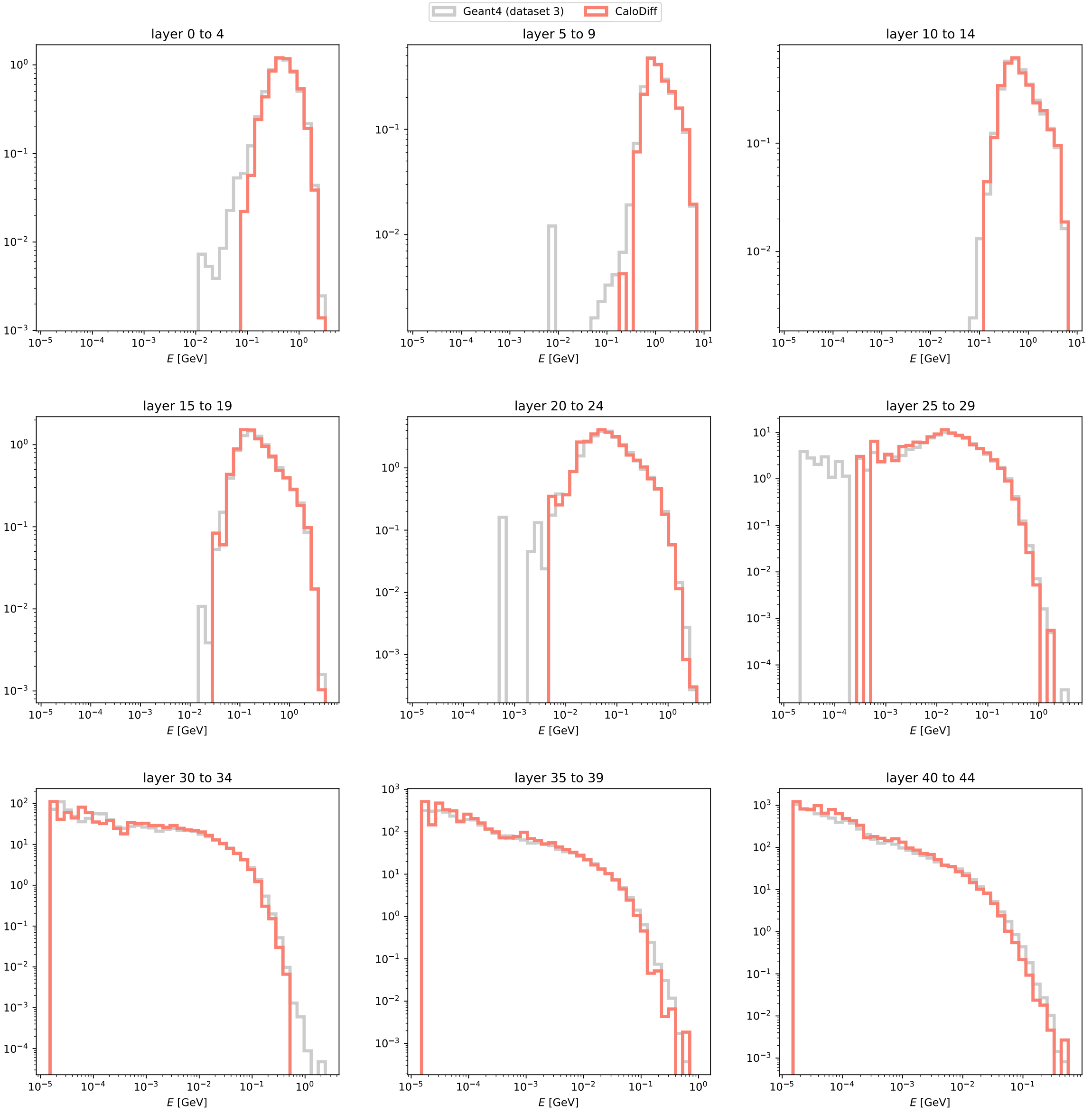


Figure 8: Layer energy (10 GeV to 100 GeV) for Dataset-3.

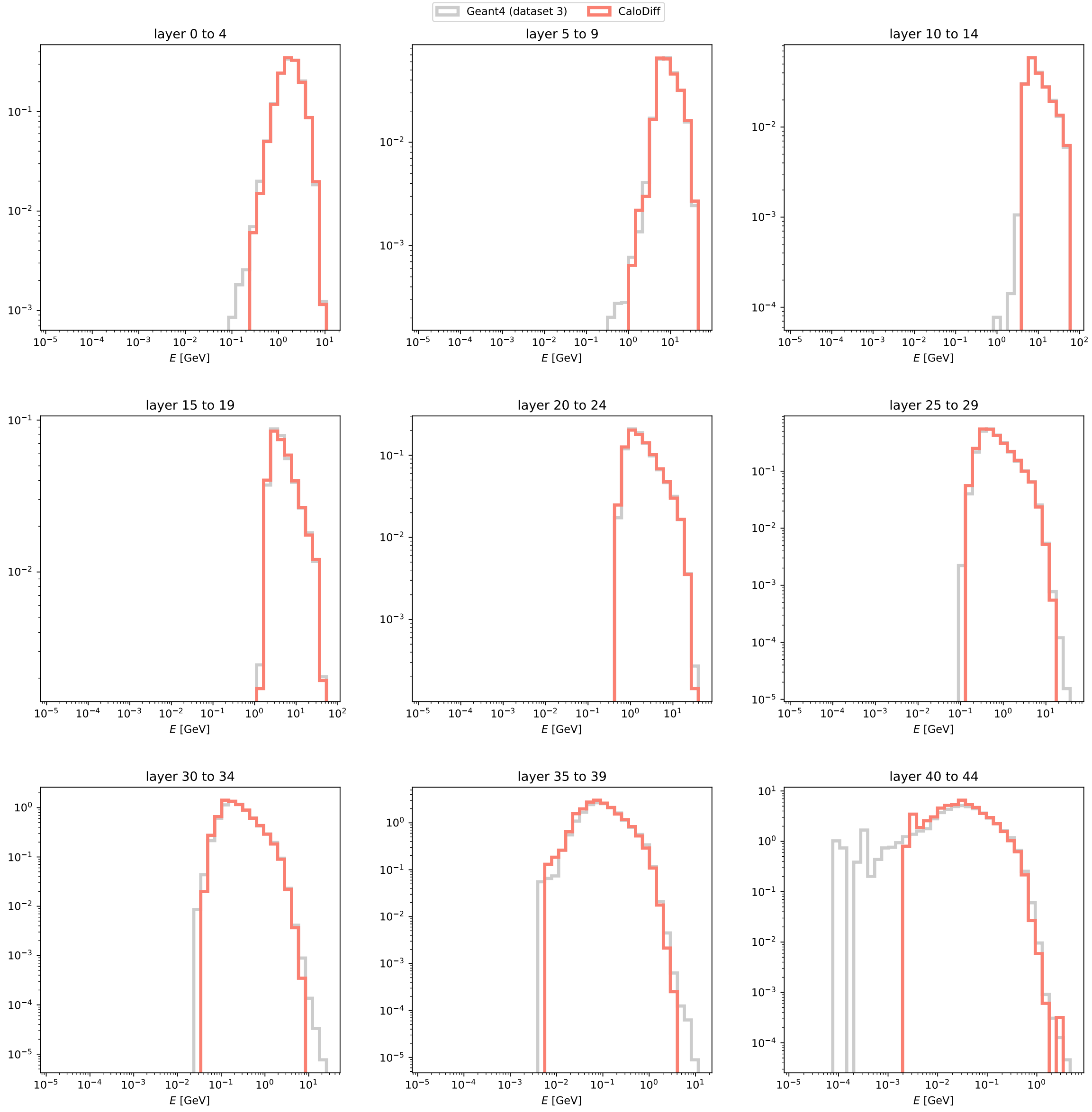


Figure 9: Layer energy (100 GeV to 1000 GeV) for Dataset-3.

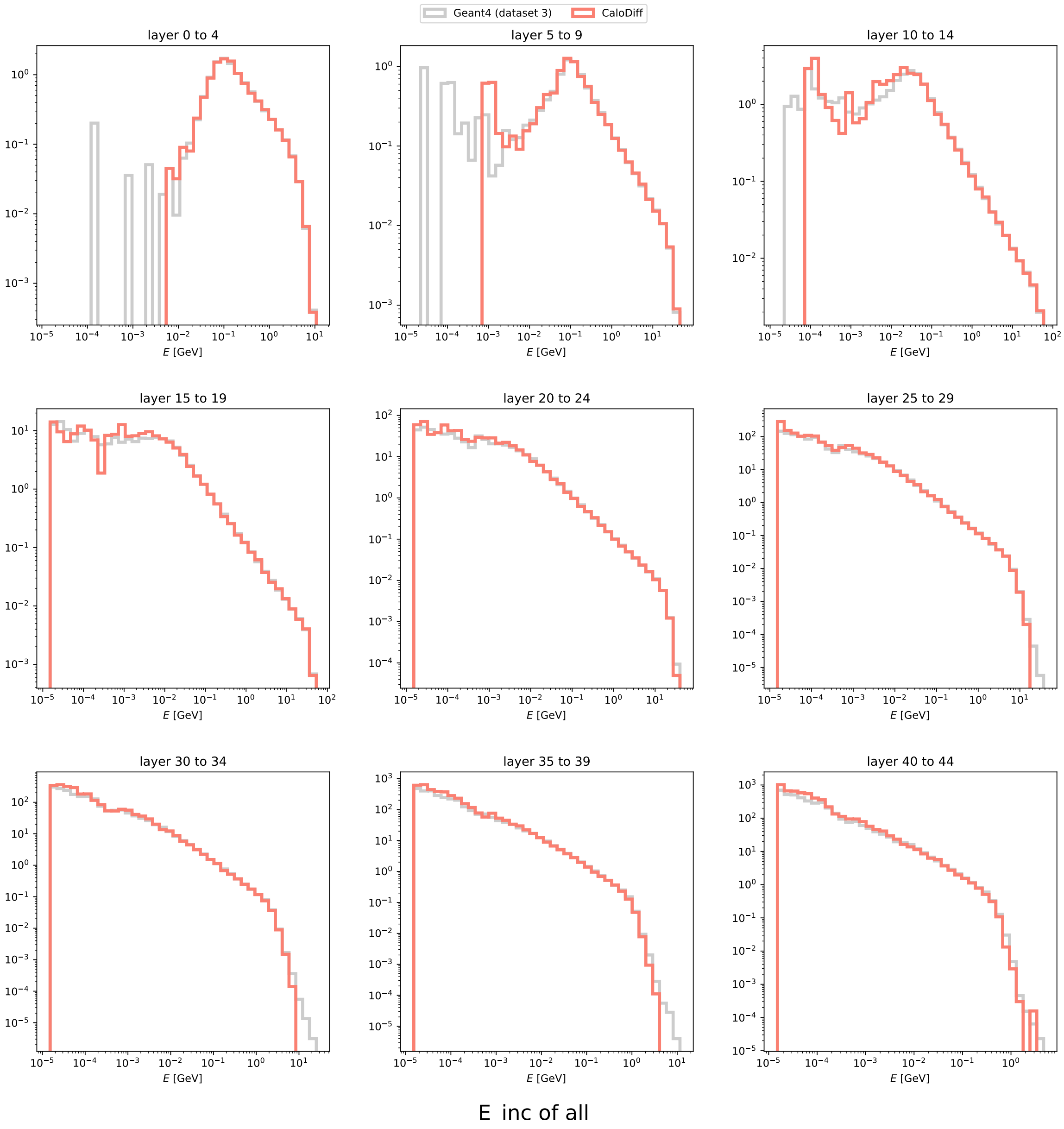
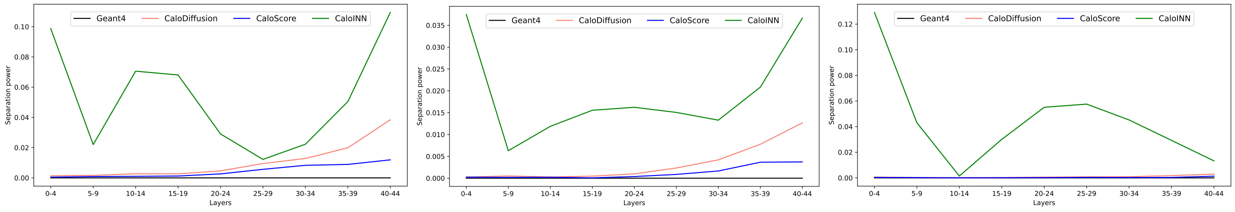
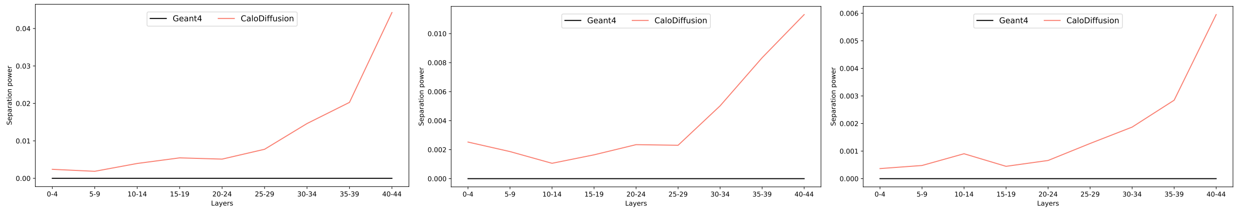


Figure 10: Layer energy (1 GeV to 1000 GeV) for Dataset-3.



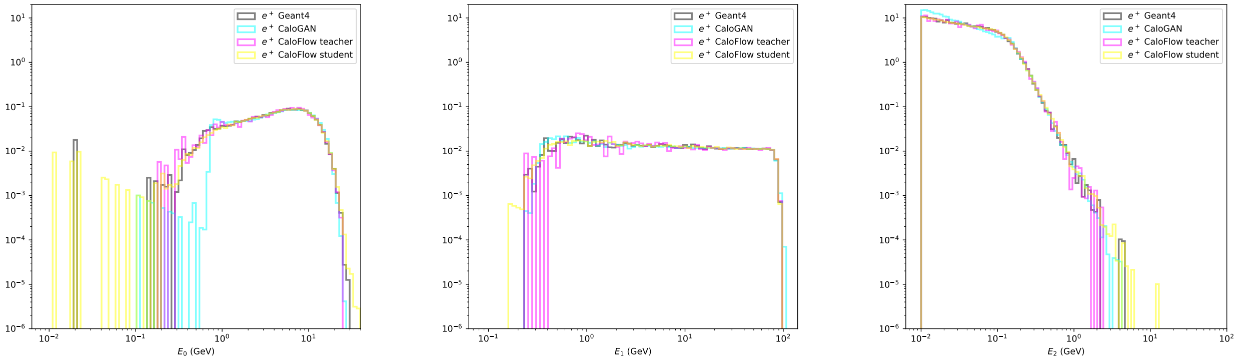
(a) Dataset-2 (1 GeV to 10 GeV). (b) Dataset-2 (10 GeV to 100 GeV). (c) Dataset-2 (100 GeV to 1000 GeV).

Figure 11: Seperation Power between Geant4, CaloDiffusion, CaloScore and CaloINN.



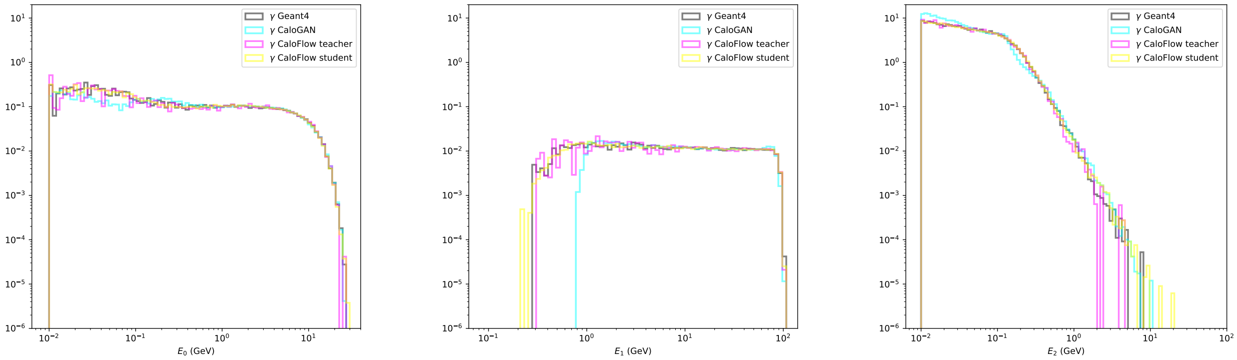
(a) Dataset-3 (1 GeV to 10 GeV). (b) Dataset-3 (10 GeV to 100 GeV). (c) Dataset-3 (100 GeV to 1000 GeV).

Figure 12: Seperation Power between Geant4 and CaloDiffusion.



(a) Layer 0 for particle eplus (b) Layer 1 for particle eplus (c) Layer 2 for particle eplus

Figure 13: Energy distribution for particle eplus in CaloGAN dataset



(a) Layer 0 for particle gamma (b) Layer 1 for particle gamma (c) Layer 2 for particle gamma

Figure 14: Energy distribution for particle gamma in CaloGAN dataset

References

[1] Geant4 in atlas, 2020.

[2] Walter Kourlitis Evangelos et al. Adelmann, Andreas Hopkins. New directions for surrogate models and differentiable programming for high energy physics detector simulation, 2022.

[3] Oz Amram and Kevin Pedro. Denoising diffusion models with geometry adaptation for high fidelity calorimeter simulation. *Physical Review D*, 108(7):072014, 2023.

[4] Matthew R Buckley, Ian Pang, David Shih, and Claudius Krause. Inductive simulation of calorimeter showers with normalizing flows. *Physical Review D*, 109(3):033006, 2024.

[5] CERN. The large hadron collider, 2024.

[6] Florian Ernst, Luigi Favaro, Claudius Krause, Tilman Plehn, and David Shih. Normalizing flows for high-dimensional detector simulations. *arXiv preprint arXiv:2312.09290*, 2023.

[7] Michele Faucci Giannelli and Rui Zhang. Caloshowergan, a generative adversarial networks model for fast calorimeter shower simulation. *arXiv preprint arXiv:2309.06515*, 2023.

[8] Sehmimul Hoque, Hao Jia, Abhishek Abhishek, Mojde Fadaie, J Quetzalcoatl Toledo-Marín, Tiago Vale, Roger G Melko, Maximilian Swiatlowski, and Wojciech T Fedorko. Caloqvae: Simulating high-energy particle-calorimeter interactions using hybrid quantum-classical generative models. *arXiv preprint arXiv:2312.03179*, 2023.

[9] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. *Advances in neural information processing systems*, 29, 2016.

[10] Claudius Krause, Ian Pang, and David Shih. Caloflow for calochallenge dataset 1. *arXiv preprint arXiv:2210.14245*, 2022.

[11] Claudius Krause and David Shih. Caloflow: fast and accurate generation of calorimeter showers with normalizing flows. *arXiv preprint arXiv:2106.05285*, 2021.

[12] Claudius Krause and David Shih. Caloflow ii: Even faster and still accurate generation of calorimeter showers with normalizing flows. *arXiv preprint arXiv:2110.11377*, 2021.

[13] Claudius Krause Ben Nachman Dalila Salamani David Shih Michele Faucci Giannelli, Gregor Kasieczka and Anna Zaborowska. Fast calorimeter simulation challenge 2022, 2022.

[14] Vinicius Mikuni and Benjamin Nachman. Score-based generative models for calorimeter shower simulation. *Physical Review D*, 106(9):092009, 2022.

[15] Vinicius Mikuni and Benjamin Nachman. Caloscore v2: single-shot calorimeter shower simulation with diffusion models. *Journal of Instrumentation*, 19(02):P02001, 2024.

[16] NVIDIA. Nvidia blackwell architecture technical brief., 2024.

[17] Michela Paganini, Luke de Oliveira, and Benjamin Nachman. Accelerating science with generative adversarial networks: an application to 3d particle showers in multilayer calorimeters. *Physical review letters*, 120(4):042003, 2018.

[18] Michela Paganini, Luke de Oliveira, and Benjamin Nachman. Calogan: Simulating 3d high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks. *Physical Review D*, 97(1):014021, 2018.

[19] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30, 2017.

[20] Henric Wilkens and (on behalf ofthe ATLAS LArg Collaboration). The atlas liquid argon calorimeter: An overview. *Journal of Physics: Conference Series*, 160(1):012043, apr 2009.