

# CALOHADRONIC: a diffusion model for the generation of hadronic showers

---

Thorsten Buss <sup>a,b</sup> Frank Gaede <sup>b</sup> Gregor Kasieczka <sup>a</sup> Anatolii Korol <sup>b</sup> Katja Krüger <sup>b</sup> Peter McKeown <sup>c</sup> and Martina Mozzanica <sup>a,1</sup>

<sup>a</sup>*Institute for Experimental Physics, Universität Hamburg*

*Luruper Chaussee 149, 22761 Hamburg, Germany*

<sup>b</sup>*Deutsches Elektronen-Synchrotron DESY,*

*Notkestr. 85, 22607 Hamburg, Germany*

<sup>c</sup>*CERN, 1211 Geneva 23, Switzerland*

*E-mail:* [martina.mozzanica@uni-hamburg.de](mailto:martina.mozzanica@uni-hamburg.de)

**ABSTRACT:** Simulating showers of particles in highly-granular calorimeters is a key frontier in the application of machine learning to particle physics. Achieving high accuracy and speed with generative machine learning models can enable them to augment traditional simulations and alleviate a major computing constraint.

Recent developments have shown how diffusion based generative shower simulation approaches that do not rely on a fixed structure, but instead generate geometry-independent point clouds, are very efficient. We present a transformer-based extension to previous architectures which were developed for simulating electromagnetic showers in the highly granular electromagnetic calorimeter of the International Large Detector, ILD. The attention mechanism now allows us to generate complex hadronic showers with more pronounced substructure across both the electromagnetic and hadronic calorimeters. This is the first time that machine learning methods are used to holistically generate showers across the electromagnetic and hadronic calorimeter in highly granular imaging calorimeter systems. The code is available at <https://github.com/FLC-QU-hep/CaloHadronic>.

**KEYWORDS:** Calorimeter methods, detector modeling and simulations

---

<sup>1</sup>Corresponding author.

---

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Dataset</b>	<b>3</b>
2.1	Data Preprocessing	4
<b>3</b>	<b>Model and Architecture</b>	<b>5</b>
3.1	POINTCOUNTFM	5
3.2	Ecal and Hcal blocks	6
<b>4</b>	<b>Results</b>	<b>10</b>
4.1	Physics Performance	11
4.2	Correlation Studies	13
4.3	Timing	15
<b>5</b>	<b>Reconstruction with Pandora Particle Flow</b>	<b>17</b>
5.1	Methods	18
5.2	Results	18
<b>6</b>	<b>Conclusion</b>	<b>19</b>
<b>7</b>	<b>Acknowledgments</b>	<b>20</b>
<b>Bibliography</b>		<b>21</b>
<b>A</b>	<b>Model details and hyper-parameters</b>	<b>27</b>
<b>B</b>	<b>Compression block</b>	<b>27</b>
<b>C</b>	<b>3D <math>\pi^+</math> showers</b>	<b>28</b>
<b>D</b>	<b>POINTCOUNTFM histograms</b>	<b>28</b>
<b>E</b>	<b>CALOHADRONIC further histograms</b>	<b>31</b>
<b>F</b>	<b>Evaluation Scores</b>	<b>32</b>
F.1	Classifier Scores	32
<b>G</b>	<b>Resolution and Linearity</b>	<b>33</b>

---

## 1 Introduction

Building generative surrogates for expensive event generation and simulation tasks is a key step in enabling the physics program of the high-luminosity LHC (HL-LHC) and future collider studies [1–3].

As experiments in high energy physics push the boundaries of luminosity resulting in ever increasing event rates, the computational demand of high-precision Monte Carlo (MC) simulations is growing to the point where it will soon surpass available computational resources [4]. Generative models offer a promising solution to this challenge, potentially reducing the immense computational load required for these simulations. This has led to substantial research into the development of machine-learning architectures tailored for more efficient and accurate detector simulation [5, 6]. Examples include generative adversarial networks (GANs) [7–18], variational autoencoders (VAEs) and their variants [18–23], normalizing flows and various types of diffusion models [23–44], as well as generative pre-trained transformer (GPT) style models [45]. The combination of a diffusion model with a transformer architecture, known as diffusion transformers [46, 47], has been used in high-energy physics for jet generation [44, 48–51].

The majority of these studies have focused on simulating electromagnetic showers. Only a few of them (e.g. [21]) have attempted to model hadronic showers, which exhibit significantly more complex and varied phenomenology with respect to electromagnetic ones. Additionally, studies have focused either on simulating showers in the electromagnetic calorimeter (ECal) or in the hadronic calorimeter (HCal) with high precision. In practice, however, a hadronic particle might already start interacting in the electromagnetic calorimeter and then continue showering in the hadronic calorimeter (and perhaps even beyond). Therefore, for realistic applications of these models, an efficient and comprehensive simulation of the entire shower is necessary. This work represents the first approach to provide a unified framework for modeling hadronic showers across both the electromagnetic and hadronic components of a highly granular calorimeter system.

As electromagnetic and hadronic calorimeters typically have different cell sizes [52] — leading to different spatial resolutions or, put differently, to a change in diameter of the shower when measured in cells (e.g. [53]) — and are built from different materials, as [54], correctly learning the behavior across the two detector systems is not trivial. Together with the much larger spatial extent and the much larger statistical fluctuations of hadronic showers compared to electromagnetic ones, this represents a major challenge for approaches based on regular grids.

The CALOHADRONIC model is a generative ML model able to generate showers in both ECal and HCal. The model uses a continuous normalizing flow (CNF) [55] for generating the number of points in each layer, coupled with EDM-diffusion models [56] for simulating showers in both calorimeters. The model uses two separate EDM-diffusion models (one for ECal and one for HCal), each designed to generate realistic showers based on their respective granularities. By using transformers within these models, CALOHADRONIC accurately generates showers in both calorimeters, respecting their distinct resolutions and materials. When generating a shower in the HCal, the ECal data is given as conditioning, and the attention mechanism captures the relevant transition between the two calorimeters.

We demonstrate the holistic generation of showers for the planned International Large Detector at the International Linear Collider. Section 2 introduces the relevant detector components and

data preparation steps. Next, Section 3 introduces the model architecture. We then first review the performance of the proposed approach at the level of simulated showers in Section 4 and also after passing through the standard reconstruction chain in Section 5. Section 6 concludes this work.

## 2 Dataset

The International Large Detector (ILD) [57] is a detector concept proposed for future  $e^+e^-$  Higgs factories, originally designed for operation at the International Linear Collider (ILC) [58]. The ILC is a proposed linear electron-positron collider that is supposed to initially operate at a center-of-mass energy of 250 GeV with the option to upgrade it to 1 TeV, with a rich program of Higgs-, Electroweak- and BSM physics [58].

The experimental community has developed designs for two complementary detectors, ILD and SiD, to optimally address the ILC goals. The emphasis for detectors at future Higgs factories is placed on ultimate precision. This requires detector technologies with new levels of performance. The momenta of the full set of final-state particles are best reconstructed with a Particle Flow Algorithm (PFA). This technique combines the information from the tracking system and from the calorimeter system to reconstruct the energy and the direction of all charged and neutral particles individually in the event. To minimize overlaps between neighboring particles, and to maximize the association accuracy between tracks and calorimeter clusters, calorimeters with very high granularity are needed. Therefore, both detector concepts employ highly granular calorimeters placed inside the solenoid coil and excellent tracking and vertexing systems. The two detector concepts differ in the choice of tracker technology. In this work, we focus on the ILD detector which would optimize the particle-flow resolution by making the detector large, thus better separating charged and neutral particles.

**ECal** Electromagnetic showers are measured with a compact highly-granular calorimeter with absorber plates made of tungsten. The ECal barrel shape is octagonal with individual stacks laid out in a way to avoid projective dead zones in azimuth. The calorimeter is composed of 30 layers. The sensitive medium consists of silicon sensors with about  $5 \times 5 \text{ mm}^2$  readout cells.

**HCal** The hadronic calorimeter consists of 48 longitudinal sampling layers containing steel absorber plates. The active layers interleaved between the absorbers feature cells of size  $3 \times 3 \text{ cm}^2$ , each comprising a scintillator tile readout individually by a silicon photomultiplier.

The ILD detector geometry is described using DD4HEP [59] version 1.30, which provides an interface to the GEANT4 toolkit [60] version 11.2.2 for simulation<sup>1</sup>, as well as the standard suite of ILD reconstruction tools [57].

**Data Generation** The dataset <sup>2</sup> used in this study was generated by firing single  $\pi^+$  particles perpendicularly to the face of the ECal.  $\pi^+$  particles were chosen, as they create hadronic showers with

---

<sup>1</sup>The QGSP\_BERT physics list is used in this study.

<sup>2</sup>A subset of 20k example showers taken from the training set is available at <https://doi.org/10.5281/zenodo.15301636>.

a distinct track like pattern resulting from ionization before the first nuclear interaction that is important to model correctly. The gun was positioned at  $(x, y, z) = (-50 \text{ mm}, 1804.7 \text{ mm}, -150 \text{ mm})$  in the ILD coordinate system. This coordinate system is defined such that the  $z$ -axis lies parallel to the beamline, the  $y$ -axis points vertically upward, and the  $x$ -axis completes the right-handed coordinate system. The particle gun was placed directly at the front surface of the ECal, so as to avoid interactions in the tracking system. The particles fly along the positive  $Y$ -axis, entering the calorimeter perpendicularly. The particle energies are uniformly distributed in the range of 10–90 GeV. A dedicated model of the ECal that has homogeneous sensitive layers without gaps between sensors was used during simulation, in order to avoid artifacts from such gaps when later placing the shower in a different position in the calorimeter.

## 2.1 Data Preprocessing

Following the approach in Ref. [30], all **GEANT4** steps are extracted from the sensitive layers of both calorimeters, yielding 2D point clouds for every one of the 78 layers. Subsequently, each layer is projected onto a virtual grid with roughly 9 times higher granularity than the original ECal and HCal:  $1.7 \times 1.7 \text{ mm}^2$  granularity is used for the ECal, while a coarser granularity of  $10 \times 10 \text{ mm}^2$  is used for the HCal. Consequently, the total number of grid points per layer is 280,900 for the ECal and 8,100 for the HCal, given a cut along the transverse plane (square cut in the  $x$ - $z$  plane) from  $-450 \text{ mm}$  to  $450 \text{ mm}$ . In a similar fashion to Ref. [30] all layers are stacked along the  $y$ -axis, with the  $y$ -coordinate assigned according to the layer number. A random offset uniformly sampled from the interval  $[-0.5, 0.5]$  is applied to each point’s  $y$ -position, dequantizing the longitudinal shower profile.

To reduce the complexity of the point cloud and the total number of points, energy depositions at the Geant4 step level below the noise cutoff threshold  $10^{-5} \text{ MeV}$  are discarded. Given the pion incident energies in the range of 10–90 GeV, the coarser granularity point cloud has an average number of points around 1 700 and a maximum number of points of 5 000. Therefore, the final shape of one shower is [5 000, 4] with four being the number of features:  $x$ ,  $y$ ,  $z$  and energy.

Standard scaling (i.e., zero mean and unit variance) is applied independently to the  $(x, y, z)$  coordinates and to the logarithm of the energy deposition,  $\log(E)$ , as well as to the incident particle energy used for conditioning. This normalization ensures more stable training of the model and mitigates the effects of scale variation.

After model inference, the individual energy depositions produced by the model are positioned into the detector geometry using the **DDML**<sup>3</sup> library [61, 62], which uses the fast simulation hooks present in **GEANT4** and **DD4HEP**. This enables an efficient placement and scoring of hits in the sensitive elements of the detector.

---

<sup>3</sup><https://github.com/key4hep/DDML/>

### 3 Model and Architecture

This section details the architecture of our proposed model, which is designed to simulate hadronic showers: CALOHADRONIC<sup>4</sup>. The model is implemented using PyTorch [63].

The CALOHADRONIC model for pion showers has three components. The first, POINTCOUNTFM, is a continuous normalizing flow [55] which is trained via flow matching [64] to learn the number of points in the highly granular grid for each layer. The second and third components are both EDM diffusion [56] models which use a transformer mechanism [46] for shower generation. The diffusion process has been implemented following [65]. One of these diffusion models (ECAL EDM-DIFFUSION) is used to generate the ECal portion of the shower, while the other (HCal EDM-DIFFUSION) generates the HCal portion.

The CNF with flow matching was chosen to generate the number of points per layer conditioned on the incident energy of the shower. Both features, points per layer in the respective calorimeters and incident energy, are then used in both the ECal and HCal diffusion models to generate the two parts of a shower. In HCal EDM-DIFFUSION, the ECal part of the shower is also provided as an additional conditioning input. Pion showers are obtained by combining the outputs of the ECAL EDM-DIFFUSION and HCAL EDM-DIFFUSION models.

#### 3.1 POINTCOUNTFM

Normalizing flows [66, 67] consist of a diffeomorphism between the physics and the latent space. The change of variables formula allows for direct likelihood-based training. However, this requires the neural network to be invertible and to have a tractable Jacobian, which imposes architectural constraints. Continuous normalizing flows [55] extend this idea by modeling the transformation as a neural ordinary differential equation (ODE) integrated over a time variable. This approach allows us to choose almost any neural network architecture. However, evaluating the likelihood now requires solving the ODE at each training step, significantly slowing down the training process.

Flow matching [64, 68] offers an efficient alternative. Instead of solving the ODE, it constructs a time-dependent velocity field. This allows training using a simple mean squared error loss, enabling training with a single forward and backward pass. Score-based diffusion models using a probability flow ODE can be seen as a special case of flow matching models.

POINTCOUNTFM is a continuous normalizing flow trained with conditional flow matching [64]. It generates the number of points deposited in each of the 78 calorimeter layers conditioned on the incident energy. For the network architecture, we use a multi-layer perceptron with five hidden layers. We apply a series of preprocessing steps to prepare the data: dequantization to handle discrete inputs, a logarithmic transformation to stretch the high-density regions, and standard scaling to normalize the features for stable training. All further hyper-parameters can be found in Table 3.

We generate new samples from the trained model with the Heun second-order ODE solver using 200 steps.

---

<sup>4</sup>The code for CALOHADRONIC including the hyperparameter settings used for training is available on <https://github.com/FLC-QU-hep/CaloHadronic>.

### 3.2 ECal and HCal blocks

Our model builds on two core ideas popularized by CaloClouds2 [69]: the use of EDM diffusion for generative modeling and representing showers as point clouds. While our approach is inspired by these principles, we propose a distinct architecture tailored to hadronic showers, introducing new components in the model structure.

Diffusion models [70] have become a powerful approach for modeling data distributions. They operate by progressively corrupting data through a forward process and then learning a reverse denoising process to recover the original distribution, effectively establishing a diffusion-based transformation between structured data and a simple prior distribution. Score-Based Continuous-Time Discrete Diffusion Models [71] extend discrete diffusion models by formulating the process in continuous time. The key difference is that they explicitly model the score function, which represents the gradient of the log-likelihood at time  $t$ . This allows for greater flexibility in sampling and training, enabling the use of stochastic differential equations (SDEs) or ordinary differential equations (ODEs) for efficient inference. They also allow custom choice of the number of steps for generation and distillation for even faster inference.

EDM diffusion [56] improves the reverse process of diffusion models by using more accurate numerical methods (higher-order solvers) that stabilize the generation process. It reduces the number of steps required for high-quality samples and speeds up inference leading to faster and more reliable diffusion sampling.

The new components in the model structure include:

**MONOTONIC WEIGHTING** It was shown in [72] that all commonly used diffusion model objectives equate to a weighted integral of evidence lower bound (ELBOs) over different noise levels, where the weighting depends on the specific objective used. In the paper it is proved that if the weighting  $w(\lambda_t)$  is monotonic, then the weighted diffusion objective of

$$\mathcal{L}_w(x) = \frac{1}{2} \mathbb{E}_{t \sim \mathcal{U}(0,1), \epsilon \sim \mathcal{N}(0,I)} \left[ w(\lambda_t) \cdot \left( -\frac{d\lambda}{dt} \cdot \|\epsilon_\theta(z_t; \lambda_t) - \epsilon\|_2^2 \right) \right]$$

is equivalent to the ELBO with data augmentation (additive noise).

The key insight is that the weighting function  $w(\lambda)$  can be chosen to be monotonic in order to give a bit more weighting to lower noise levels, which is where the model is most accurate.

Inspired by the non-monotonic EDM weighting function of EDM diffusion [56], they evaluate a variant referred to as *EDM-monotonic*. This function matches the original EDM weighting  $\tilde{w}(\lambda)$ , except it is made monotonic by setting  $w(\lambda) = \max_\lambda \tilde{w}(\lambda)$  for all  $\lambda < \arg \max_\lambda \tilde{w}(\lambda)$ .

**FOURIER LAYER** Ref. [73] found empirical evidence of a spectral bias: i.e. lower frequencies are learned first in a neural network and learning higher frequencies gets easier with increasing manifold complexity. They show that mapping the inputs to a higher dimensional space using high frequency functions before passing them to the network enables better fitting of data that contains high frequency variation. Ref. [74] then leveraged these findings in the context of neural scene representations.

In the same way, one could enhance the sensitivity of the model to higher frequency by manually

embedding the input. This is done by applying a composition of two functions to the input. Using the same formalism as [74], we can use the composition  $F_\Theta \circ \lambda$  where  $F_\Theta$  is the learned while  $\lambda$  is not.  $F_\Theta$  is a regular MLP while  $\lambda$  is a mapping  $\mathbb{R} \rightarrow \mathbb{R}^{2L}$  defined by:

$$\lambda(p) = [\sin(2^0\pi p), \cos(2^0\pi p), \dots, \sin(2^{L-1}\pi p), \cos(2^{L-1}\pi p)]. \quad (3.1)$$

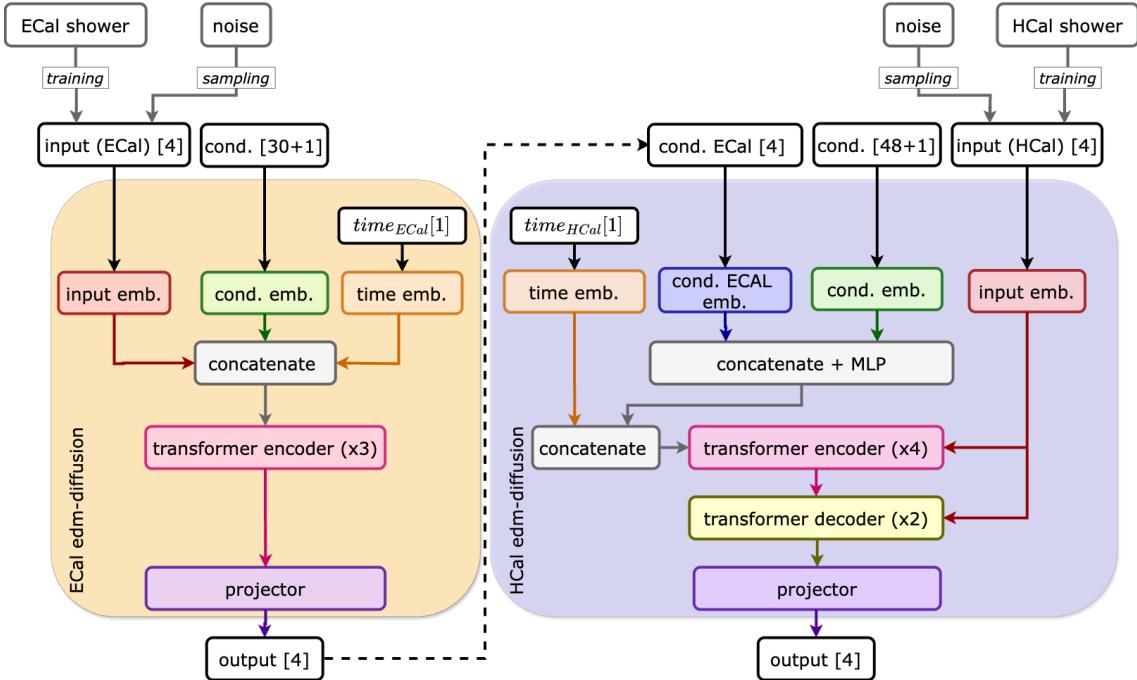
In our case L is set to 10.

**ATTENTION** Hadronic showers have multiple differences with respect to electromagnetic showers. One of them is the common appearance of tracks within the shower. These tracks are due to charged secondary particles traversing multiple calorimeter layers without undergoing an inelastic collision or being stopped. In the previous work of CaloClouds II [69], the authors used a point-wise approach to model electromagnetic (EM) showers. Since EM showers are topologically more uniform and isotropic (typically with an ellipsoid-like shape), the point-wise layers were sufficient to model the structure of EM showers while adding very little computational overhead. However in the case of  $\pi^+$  showers, the point-wise approach fails to capture complex interactions between points, such as tracks, as it treats each point independently. This is problematic for hadronic showers, where modeling the fine structure of the shower is crucial. The attention mechanism, and consequently the transformer architecture [46], offers a potential solution to this problem. Attention allows the relationships between different points at different positions in the shower to be modeled, capturing the underlying structure and improving the fidelity of generated hadronic shower events.

An illustration of the two blocks that compose the model — ECAL EDM-DIFFUSION and HCAL EDM-DIFFUSION — is shown in figure 1. Both blocks are based on EDM diffusion.

**ECAL EDM-DIFFUSION** An illustration of ECAL EDM-DIFFUSION is shown on the left of figure 1. For ECAL EDM-DIFFUSION, the conditional features are the incident energy and the number of points per layer for each of the 30 layers of the electromagnetic calorimeter of ILD. In their embedding, both of them are concatenated and then passed through two fully connected layers. The time feature of the diffusion model also has an embedding, consisting of a Fourier feature mapping and then two fully connected layers. The Fourier mapping of the time variable is taken from [56] and is similar to the previously mentioned Fourier Layer. The 4D input, consisting of the three coordinates ( $x, y$  and  $z$ ) and the energy ( $e$ ) of the input shower, goes into custom defined embedding layers based on the Fourier Layer and then two fully connected layers as well. Time, conditioning and input embeddings, with a dimension of 128, are concatenated along the points axis and fed into three layers of transformer encoder [46]. After removing the conditioning points, a projection with three fully connected layers is applied to map to the desired output dimension of four features:  $x, y, z$ , *energy*.

**HCAL EDM-DIFFUSION** An illustration of HCAL EDM-DIFFUSION is shown in figure 1 on the right. In HCAL EDM-DIFFUSION the conditional features are the incident energy, the number of points per layer and the ECal shower (all points). The hadronic calorimeter of ILD consists of 48 layers. The time feature of the diffusion model has an identical embedding as explained in ECAL EDM-DIFFUSION. This is also true of the incident energy and the number of points per layer, with the only exception being the embedding dimension 64 instead of 128. The ECal conditioning is first passed



**Figure 1.** Illustration of the structure of the ECAL EDM-DIFFUSION (left) and HCAL EDM-DIFFUSION (right). *time* is the time of the diffusion model used to add noise to the data and *cond.* are the incident energy and the number of points per layer. The electromagnetic calorimeter (ECal) consists of 30 layers, and the hadronic calorimeter (HCal) consists of 48 layers. *input (ECal or HCal)* is the input shower to be learned during training.

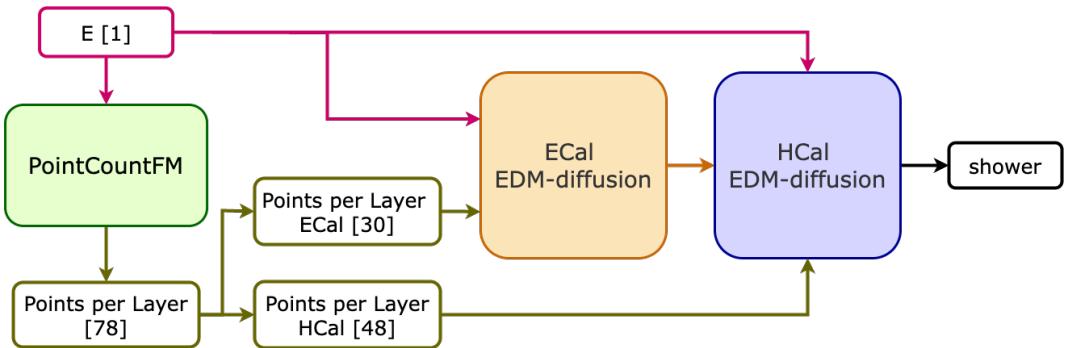
into a set compression layer. Drawing inspiration from [75], this layer maps the point cloud data to a fixed-size set of latent tokens, 10 in this case. More information on this layer can be found in appendix B. The ECal showers are then fed into two fully connected layers, concatenated together with the conditioning vector along the points axis and then passed through one linear layer to match the model dimension of 128. At this point it is concatenated with the time feature.

The input of the model, which is the HCal part of the pion shower, has the same embedding as the input of ECAL EDM-DIFFUSION 3.2. Both the input and the conditional features are fed into four layers of transformer encoder [46]. Its output is then concatenated with the input and passed through two layers of transformer decoder [46]. Finally, a projection consisting of three fully connected layers is applied to produce the desired output dimensionality.

**Training and Sampling** During training a random continuous time step is added to the input and the model is trained to denoise the data. The EDM diffusion models are trained conditioned on the shower energy and number of points per layer. The loss is approximated by a simple mean squared error (MSE) between a rescaled version of the input data and the denoised output.

During sampling, the conditional POINTCOUNTFM generates the number of points per layer for a given incident energy. The EDM diffusion blocks start from gaussian noise and use a stochastic sampler that combines a 2nd order deterministic ODE integrator with explicit Langevin-like “churn”

— a process of periodically adding and removing noise to encourage exploration and improve sample diversity. During sampling, the inputs in figure 1 start as Gaussian noise and the points are then moved step-by-step to create a shower. The Number of Function Evaluations (NFE) refers to the number of times a neural network that is used to model the reverse diffusion process needs to be evaluated during the sampling phase to produce a final sample. Since CALOHADRONIC is a continuous time score-based diffusion model, one has the flexibility to choose the NFE by finding a good compromise between precision and speed-up. In CALOHADRONIC 59 NFE were used. An illustration of shower generation is shown in figure 2.



**Figure 2.** Illustration of the generation process of CALOHADRONIC. The POINTCOUNTFM generates the number of points per layer. The ECal points per layer distributions along with E are fed into the ECAL EDM-DIFFUSION block. Its output, E and the HCal points per layer distributions are passed into the HCAL EDM-DIFFUSION block. By concatenating the output of the two blocks a new pion shower is generated.

The new components in the model training process include:

**OPTIMIZER** The Adam [76] optimizer is a popular choice for training deep learning models due to its adaptive learning rate and momentum features. However, it can be memory-intensive, especially when training large models such as diffusion models. In [77], the authors propose a new optimizer called Adam-mini, which is a variant of Adam that uses a smaller memory footprint and is more efficient for training diffusion models. The idea for Adam-mini can be understood by delving into the Hessian structure of neural networks and recalling the fact that it is a near-block diagonal matrix. In [77], the authors find that, for each of these dense sub-blocks, there exists a single high-quality learning rate that outperforms Adam. The memory footprint can be reduced by assigning a single learning rate to each of these dense sub-blocks instead of a separate learning rate per parameter as in Adam.

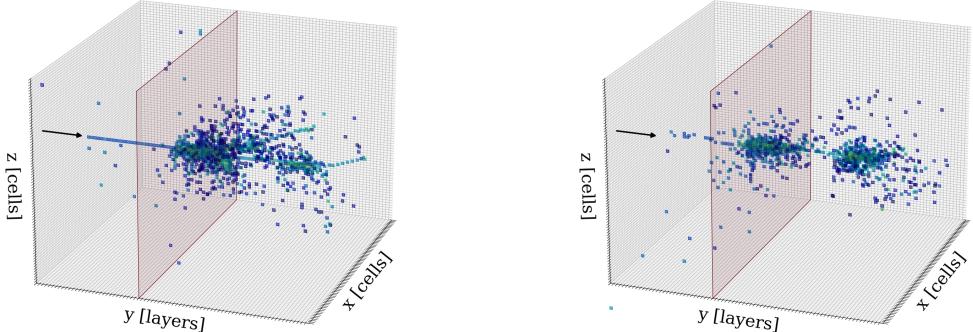
**SCHEDULER** Drawing inspiration from [78], the learning rate scheduler has been switched from using a constant learning rate with linear decay to OneCycleLR to improve training stability and convergence. OneCycleLR starts with a linear warm-up, allowing the model to gradually increase the learning rate, which helps stabilize the initial phase of training. After the warm-up, the scheduler applies cosine annealing decay, enabling faster convergence and avoiding premature reduction of the learning rate during fine-tuning. In [79] [80] it has been shown that OneCycleLR accelerates

convergence and improves generalization, making it particularly effective for complex models like transformers and diffusion models.

## 4 Results

In the following, the results of generating  $\pi^+$  showers in the electromagnetic and hadronic calorimeter are presented. After the sampling procedure explained in 3.2, the effects of preprocessing were reversed during post-processing and a calibration on the number of points per layer was applied. Although not strictly necessary —since the model also predicts the  $y$ -axis— this calibration is used because the POINTCOUNTFM model better captures the distribution of points per layer, whereas the diffusion model can result in a less precise approximation along the layer ( $y$ ) coordinate. The calibration was done by considering the number of points per layer computed by POINTCOUNTFM and then assigning the generated points by CALOHADRONIC to the corresponding layer.

The final point cloud showers are projected back to a regular grid in which the cell size is the same as that of the ILD calorimeters, hence  $5 \times 5 \text{ mm}^2$  for the ECal and  $30 \times 30 \text{ mm}^2$  for the HCal. For downstream analyses a cell energy cut at  $\sim 10^{-2} \text{ MeV}$  is applied, since below this threshold the sensor response is indistinguishable from electronic noise. This cut was applied to all showers when calculating the shower observables and scores in this section. A first naive comparison of the showers generated by CALOHADRONIC with Geant4 simulated  $\pi^+$  showers can be made from a visual inspection of whether the final projection of the showers resembles the Geant4 equivalent. In figure 3 the 3D view of a 50 GeV Geant4 shower (left) and a 50 GeV CALOHADRONIC shower (right) are shown.



**Figure 3.** 3D view of a 50 GeV  $\pi^+$  shower simulated with Geant4 (left) and a 50 GeV shower generated with CALOHADRONIC (right). The color represents the energy deposition in the cells. The red plane represents the division between ECal and HCal at layer 30.

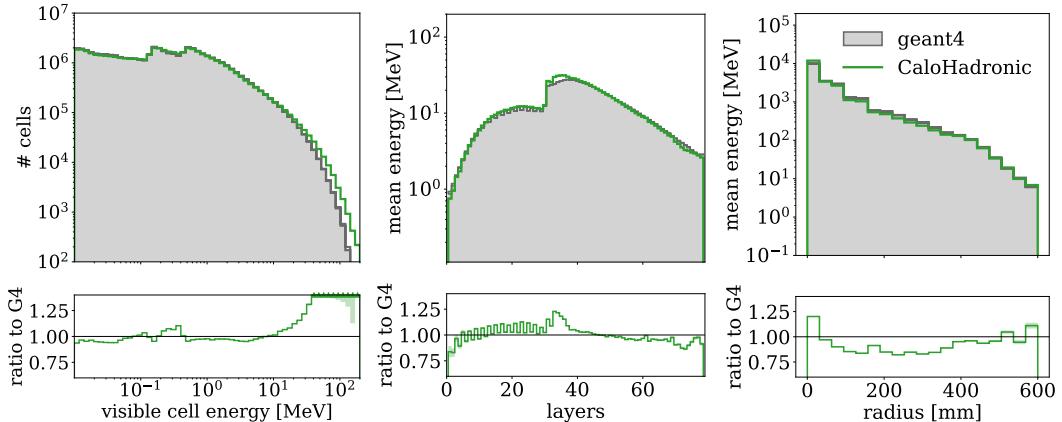
Figure 3 shows that the tracks resulting from hadronic interactions occurring in  $\pi^+$  showers, are well reproduced by CALOHADRONIC. These was possible due to the self-attention mechanism

that took care of the interactions among points. The shower core is well defined and the tracks are visible. Notably, the CaloHadronic model better reproduces fine-grained, track-like structures in the HCal, which are especially evident when compared to the Geant4 reference. Noise manifests as low-energy deposits spread outside the core of the shower and can arise from detector effects such as electronics noise, cross-talk, or background activity. More generated  $\pi^+$  showers can be found in appendix D.

#### 4.1 Physics Performance

In this section, various calorimeter shower distributions for both the Geant4 test set and datasets generated using CALOHADRONIC are compared. In figures 4 and 5 cell-level distributions are shown for 50k  $\pi^+$  showers across both the ECal and HCal with an incident energy uniformly distributed between 10-90 GeV.

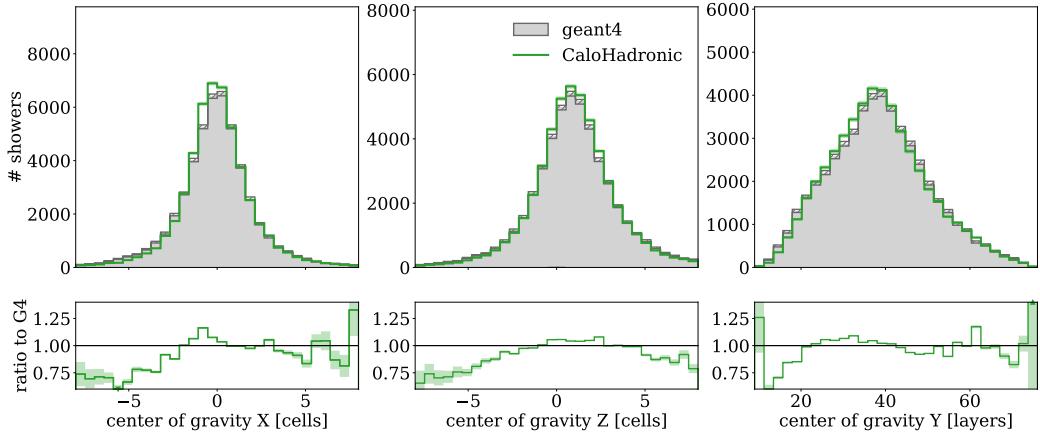
In figure 4 the energy distribution per cell (left), as well as the longitudinal (center) and radial (right) shower profiles are shown.



**Figure 4.** Histogram of the cell energies (left), longitudinal shower profile (center), and radial shower profile (right) for Geant4 and CALOHADRONIC. All distributions are calculated with 50000 events sampled with a uniform distribution of incident particle energies between 10 and 90 GeV. The bottom panel provides the ratio to Geant4. The error band corresponds to the statistical uncertainty in each bin.

The per-cell energy distribution (figure 4, left) contains the energy of the cells of all showers in the test dataset. In this distribution there are two peaks. The first one at  $\sim 0.2$  MeV corresponds to the most probable energy deposition of a minimum ionizing particle (MIP) in the silicon sensors of the ECal while the second one at  $\sim 1$  MeV corresponds to the MIP of the HCal. The model describes this distribution reasonably well. The largest discrepancy appears for higher cell energies where the statistics are lower.

The longitudinal shower profile (figure 4, center) describes how much energy is deposited on average in each of the 78 calorimeter layers. The first 30 layers are part of the ECal, while the last 48 compose the HCal. The model describes the distribution well. It can be observed that there are small bumps in the ECal layers, where the Geant4 simulation differs between even and odd layers.



**Figure 5.** Position of the center of gravity of showers along the  $x$  (left),  $z$  (center), and  $y$  (right) directions. All distributions are calculated for 50000 showers with a uniform distribution of incident particle energies between 10 and 90 GeV. The error band corresponds to the statistical uncertainty in each bin.

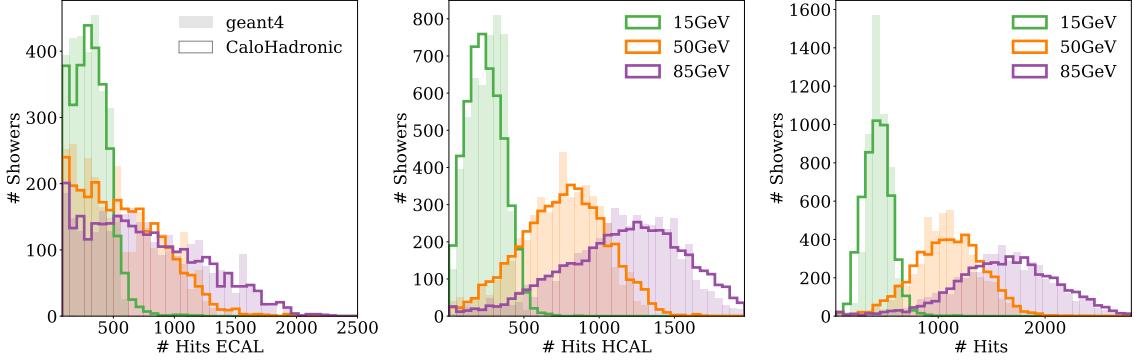
This alternating structure of higher and lower energy depositions per layer arises from the fact that for technical reasons, pairs of silicon sensors are placed on either side of one tungsten absorber layer facing opposite directions. These slabs are then installed into a tungsten structure containing every other absorber layer. The different amount of nonsensitive material results in the observed pair-wise difference in the mean energy between consecutive layers. CALOHADRONIC has some difficulties in fully modeling this difference.

The radial shower profile (figure 4, right) describes the average radial energy distribution around the central shower axis (in the  $y$ -direction) in both ECal and HCal. CALOHADRONIC has a tendency to generate slightly narrower showers than Geant4.

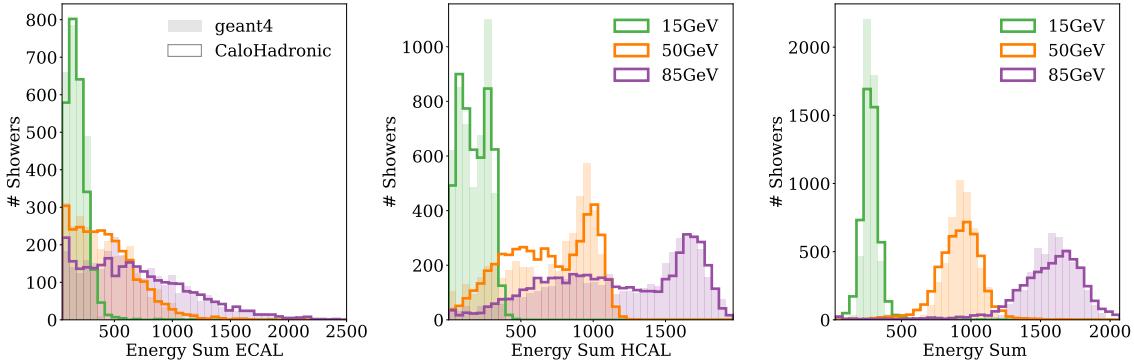
In figure 5 distributions for the center of gravity (the energy weighted shower centroid, CoG) in the  $x$ ,  $z$  and  $y$  directions are shown. CALOHADRONIC models the CoG in the  $x$ ,  $z$  and  $y$  directions relatively well. The CoG in  $x$  and  $z$  are slightly too narrow, while the CoG along  $y$  is slightly shifted to the left. It should be noted that the center of gravity along the incident direction ( $y$ ) is also affected by the points per layer calibration of POINTCOUNTFM.

We now investigate the performance of CALOHADRONIC at specific incident energies. Given that the dataset contains energies ranging from 10 to 90 GeV, incident energies of 15 GeV, 50 GeV, and 85 GeV were selected to represent the lower, middle, and upper ends of the range, respectively. The Geant4 incident energies are selected by extracting 5,000 showers from the initial dataset within a  $\pm 1$  MeV window centered on the chosen incident energies. The CALOHADRONIC showers, in contrast, are generated anew using the same incident energies as those used for Geant4.

In figure 6 the number of hits for 5k showers at three incident energies is shown for the ECal and HCal individually, as well as for both calorimeters combined. The number of hits is defined as the



**Figure 6.** Number of hits per shower at incident energies of 15 GeV (green), 50 GeV (orange), and 85 GeV (purple) for ECal (left), HCal (center), and for both calorimeters combined (right). For each energy, 5,000 showers are shown. Geant4 showers are sampled within  $\pm 1$  MeV of the target energy; CALOHadronic showers are newly generated at the same incident energies.



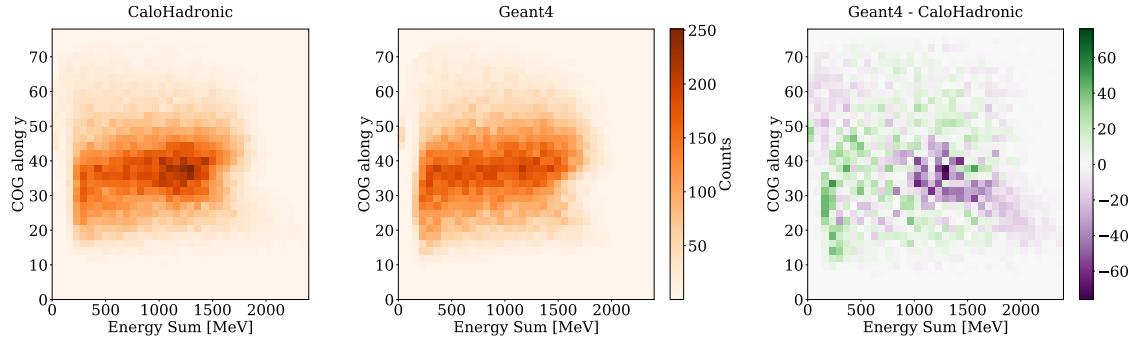
**Figure 7.** Total visible energy per shower at incident energies of 15 GeV (green), 50 GeV (orange), and 85 GeV (purple) for ECal (left), HCal (center), and for both calorimeters combined (right). For each energy, 5 000 showers are shown. Geant4 showers are sampled within  $\pm 1$  MeV of the target energy; CALOHadronic showers are newly generated at the same incident energies.

number of cells with energy above the threshold of 0.02 MeV. In figure 7 the energy sum (total visible energy) for 5k showers at three incident energies for ECal, HCal and both calorimeters combined is shown. The total energy and the number of hits are represented well by CALOHadronic. In appendix G both resolution and linearity plots are also shown.

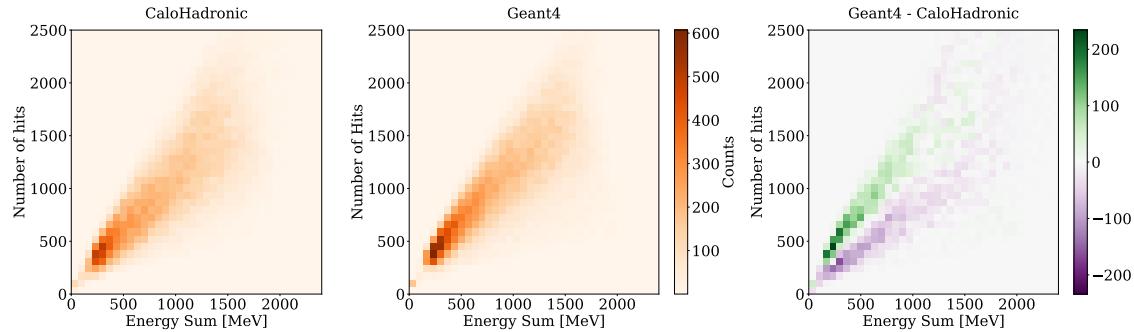
## 4.2 Correlation Studies

In this section, 2D histograms between the shower-level observables described in 4.1 are studied. These histograms enable an investigation into how well some correlations are learned by CALOHadronic. Investigations are performed considering both the entire incident energy range (10-90 GeV) and for single incident energies.

In figure 8 2D histograms of the center of gravity versus the energy sum per shower are shown for the incident energy range of 10-90 GeV for 50k showers. The difference between the two 2D



**Figure 8.** 2D histograms of the center of gravity in the  $y$  direction and the energy sum per shower for 50k showers with incident energies in the range of 10-90 GeV. The left plot shows the CALOHADRONIC distribution, the center plot shows the Geant4 distribution. The color represents the number of showers. The right histogram shows the difference between the two.

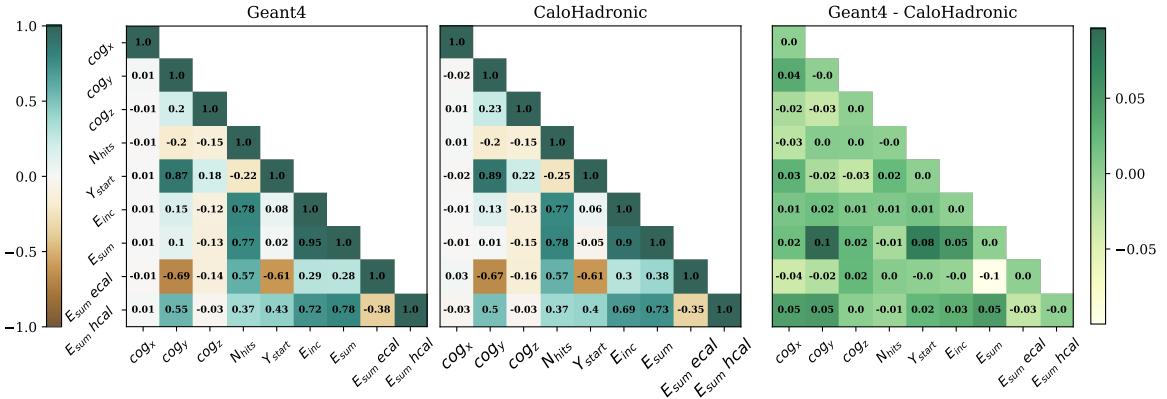


**Figure 9.** 2D histograms of the number of hits and the energy sum per shower for 50k showers with incident energies in the range of 10-90 GeV. The left plot shows the CALOHADRONIC distribution, the center plot shows the Geant4 distribution. The color represents the number of showers. The right histogram shows the difference between the two.

histograms is also shown, in order to highlight regions with significant deviations while accounting for statistical fluctuations.

In figure 9 2D histograms between the number of hits and the energy sum per shower, as well as their difference, are shown for the incident energy range of 10-90 GeV for 50k showers.

In figures 8 and 9 both histograms show good agreement between CALOHADRONIC and Geant4. Deviations are visible if one looks at the difference. In figure 8 the region where the CoG along  $y$  is between the 30th and 50th layer exhibits the largest discrepancies between Geant4 and CALOHADRONIC. Within this range, CALOHADRONIC tends to over-populate the central part of the energy sum distribution while under-populating the tails. The difference in the remaining regions of the histogram can be considered negligible, as the statistics are relatively low (fewer than 50 showers per bin). In figure 9, the orange 2D histograms show a good agreement between Geant4 and CALOHADRONIC distributions. However, the difference plot reveals a systematic shift. For a fixed number of hits, CALOHADRONIC shows an excess of events at lower energy sums and a deficit at higher ones, compared to Geant4. On the other hand, for a fixed energy sum, CALOHADRONIC



**Figure 10.** Pearson correlation coefficient matrix for key shower observables for both Geant4 (left) and CALOHADRONIC (center). The rightmost panel displays the difference between the two matrices.

has fewer events at lower hit counts and more at higher hit counts.

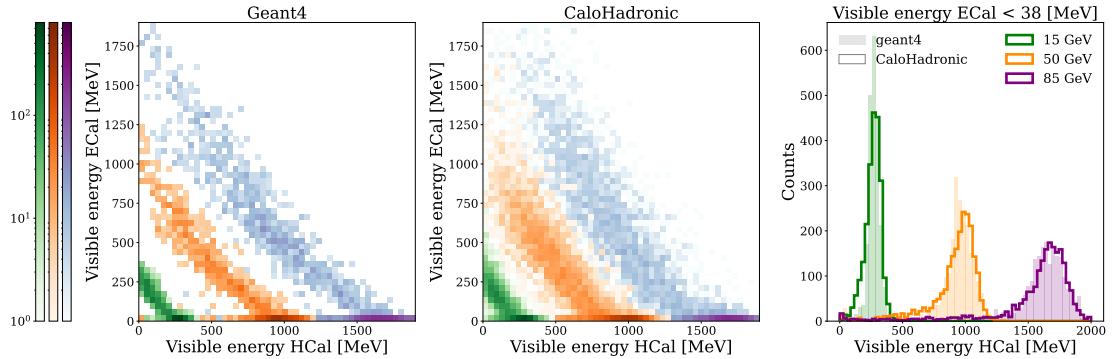
To investigate the correlation aspects of  $\pi^+$  shower generation in greater depth, one can compute the Pearson correlation coefficient between relevant observables, such as the three center of gravity observables, incident energy, shower start layer, total visible energy and total visible energy in the ECal and HCal separately per shower. In the absence of precise timing information, we define the "shower start layer" as the layer containing the most energetic cell. While this does not guarantee that the shower physically starts in this layer, it provides a practical proxy based on the assumption that the initial stages of the shower typically deposit the largest energy near the start. This coefficient quantifies the linear relationship between two variables, providing a value between -1 and 1, where 1 indicates a perfect positive correlation, -1 a perfect negative correlation, and 0 no linear correlation. The table for the Pearson correlation coefficient for Geant4 and CALOHADRONIC is presented in figure 10 along with their difference.

Figure 10 shows a good agreement between the Pearson correlation coefficient of Geant4 and CALOHADRONIC. As one can see from the rightmost table, the variables in which CALOHADRONIC coefficients vary most significantly with respect to Geant4 (Pearson coefficient difference  $> 0.07$ ) are the ones with total visible energy versus the center of gravity along y, shower start layer, and the total visible energy deposited in the ECal.

Figure 11 shows two 2D histograms of the total visible energy deposited in the ECal against the total visible energy deposited in the HCal for incident energies of 15 GeV, 50 GeV and 85 GeV, respectively. The 1D histogram on the right side gives a clearer view of the HCal total visible energy when the ECal total visible energy is below 38 MeV. Most of the pion showers lie in the region of lower energy depositions in the ECal and higher energy depositions in the HCal. This is due to the fact that a large fraction of the pions travel through the ECal before showering in the HCal, where the majority of their energy is deposited.

### 4.3 Timing

The primary objective of training generative networks on calorimeter showers is to achieve a significant speedup compared to Monte Carlo-based simulators. To evaluate the performance gain,



**Figure 11.** In the two 2D histograms the energy sum in the electromagnetic calorimeter and the energy sum in the hadronic calorimeter is shown for 15 GeV (green), 50 GeV (orange) and 85 GeV (purple) showers for both Geant4 (left) and CALOHADRONIC (center). The color represents the number of showers. Here 5,000 showers were used. Geant4 showers are sampled within  $\pm 1$  MeV of the target energy; CALOHADRONIC showers are newly generated at the same incident energies. The 1D histogram (on the right) shows the HCal energy sum when the ECal energy sum is below 38 MeV.

we conduct a timing study to compare the shower generation speed of CALOHADRONIC with that of Geant4. The times are computed from three runs, each using incident energies ranging from 10 to 90 GeV in 10 GeV increments, with 100 showers per energy point.

The timing evaluation shown in Table 1 presents results for both a single AMD EPYC 7402 CPU and an NVIDIA® A100 GPU 80GB. To ensure a fair comparison with Geant4 for the CPU timing, only a single core of the CPU is used. The mean and standard deviation of the time per shower over 3 runs for both GPU and CPU were calculated. The results are presented for various number of function evaluations (NFE) in the generation and for different batch sizes. Batch sizes of 1 and 16 were chosen because initial studies showed that in order to generate all the showers, batch sizes of 16, 32 and 64 performed similarly, while a batch size of 128 was slower. This is likely due to the fact that for batch sizes which are too large the GPU cash overflows, resulting in a slowdown.

In table 1 it can be seen that CALOHADRONIC is faster than Geant4 on CPU, and is significantly faster on GPU, for a single forward evaluation. As soon as the number of function evaluations (NFE) increases, i.e. 29 or 59, the model is slower than Geant4 on CPU while on GPU it is still faster. Following [56], each Heun step requires two function evaluations, except for the final iteration, where a first-order Euler step is used instead of the second-order Heun method, resulting in only one function evaluation at the end. A second study of how each of the three components of CALOHADRONIC contributes to the overall generation time was performed. Around 80% is due to HCAL EDM-DIFFUSION, around 18% to ECAL EDM-DIFFUSION and around 2% to POINTCOUNTFM. This is due to the fact that HCAL EDM-DIFFUSION and ECAL EDM-DIFFUSION are both based on transformers which are known to be computationally expensive. Here the advantages of a smaller model for ECAL EDM-DIFFUSION with respect to HCAL EDM-DIFFUSION are clear.

For both ECAL EDM-DIFFUSION and HCAL EDM-DIFFUSION models, sampling was performed using 59 NFE. As a result, generating each event requires multiple forward evaluations of the model, which directly impacts inference speed. This computational cost highlights a key avenue for future improvement—namely, the reduction of the NFE through the use of accelerated samplers such as

<b>Simulator</b>	<b>Hardware</b>	<b>NFE</b>	<b>Batch Size</b>	<b>Time / Shower [s]</b>	<b>Speed-up</b>
Geant4	CPU		1	$2.09 \pm 0.05$	$\times 1$
CALOHADRONIC	CPU	1	1	$0.591 \pm 0.001$	$\times 3.5$
			16	$0.7342 \pm 0.0006$	$\times 2.8$
		29	1	$17.2 \pm 0.1$	-
			16	$21.37 \pm 0.05$	-
		59	1	$34.8 \pm 0.4$	-
			16	$43.3 \pm 0.3$	-
	GPU	1	1	$0.0086 \pm 0.0007$	$\times 243$
			16	$0.0033 \pm 0.0009$	$\times 633$
		29	1	$0.1978 \pm 0.0007$	$\times 11$
			16	$0.0752 \pm 0.0004$	$\times 28$
	GPU	59	1	$0.3962 \pm 0.0008$	$\times 5$
			16	$0.1531 \pm 0.0002$	$\times 14$

**Table 1.** Comparison of the computational performance of CALOHADRONIC to the baseline Geant4 simulator. All CPU runs were performed on a single core of an AMD EPYC 7402 CPU, and GPU runs used an NVIDIA® A100 with 80 GB memory. For each configuration the showers are generated using incident energies ranging from 10 to 90 GeV in 10 GeV increments, with 100 showers per energy point. The mean and standard deviation is computed across three runs. Results show mean  $\pm$  std over the 3 runs.

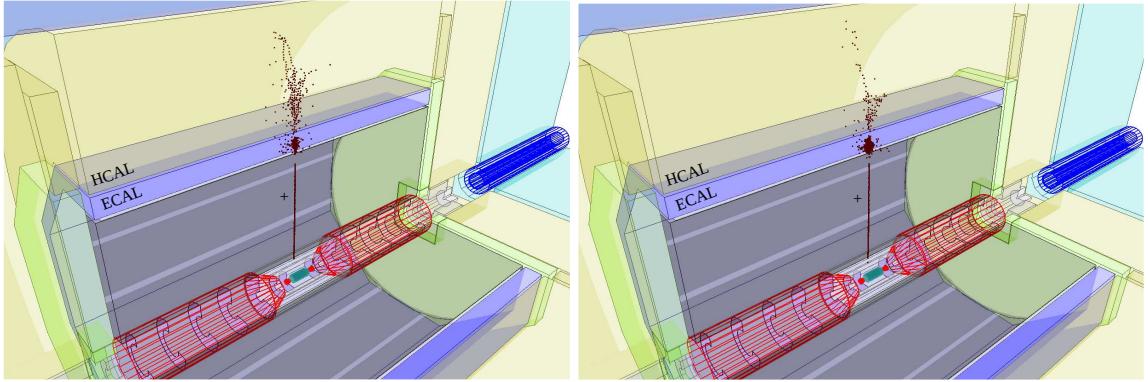
consistency models or distillation techniques.

## 5 Reconstruction with Pandora Particle Flow

For a fast calorimeter shower model to be used to produce simulated data for physics analyses, it must ultimately be interfaced with the reconstruction chain used by a given experiment. Detectors designed for operation at future  $e^+e^-$  colliders are typically optimized for the particle flow approach to reconstruction. The goal of this approach is to individually reconstruct each particle present in an event. To this end, measurements of charged particles are made in the tracking system, while measurements of neutral particles are made in the calorimeters. This drives the need for high granularity in the calorimeter systems, which is essential to correctly separate clusters of hits associated with neutral particles from those associated with charged particles.

In this study, we employ the state-of-the-art PANDORAPFA particle flow algorithm [81, 82], which is used in the ILD standard reconstruction chain. The simulated hits produced by either Geant4 or the generative model first undergo a digitization procedure, which takes into account detector effects such as noise from the electronics. A two-step calibration procedure is then applied to the hits, after which the sum off all the hits in the shower corresponds to the energy of the incident particle. It is these digitized/calibrated calorimeter hits, together with tracks formed in the tracking system which are provided as input to PANDORAPFA.

PANDORAPFA then applies a complex series of pattern recognition algorithms, which ultimately aim to correctly form calorimeter clusters and assign corresponding tracks where appropriate. The



**Figure 12.** Event displays of Pandora PFOs reconstructed from a Geant4 shower (left), and a CALOHADRONIC shower (right).

output produced by PANDORAPFA is a list of reconstructed objects referred to as *Particle Flow Objects* (PFOs), each of which contains information about the particle’s four-momentum and ID.

### 5.1 Methods

In order to be able to apply the standard reconstruction suite of ILD to showers produced by the model, it is necessary to be able to correctly place the energy deposits into the detector geometry. We therefore use the DDML library [61, 62] to combine the model output with a full simulation application. We fire a  $\pi^+$  with an energy of 50 GeV from a particle gun positioned at  $(-5, 0, -15)$  mm<sup>5</sup> in the global ILD coordinate system, with a direction chosen to produce a perpendicular incidence at the calorimeter face. This takes into account the slight curvature of the track. The particles are not produced directly at the face of the calorimeter, as it is necessary for a track to be present in the tracking system for PANDORAPFA to apply the correct sequence of algorithms for  $\pi^+$  reconstruction. However, the energy of the pion used for conditioning of the model is extracted directly at the face of the calorimeter, to match the training scenario.

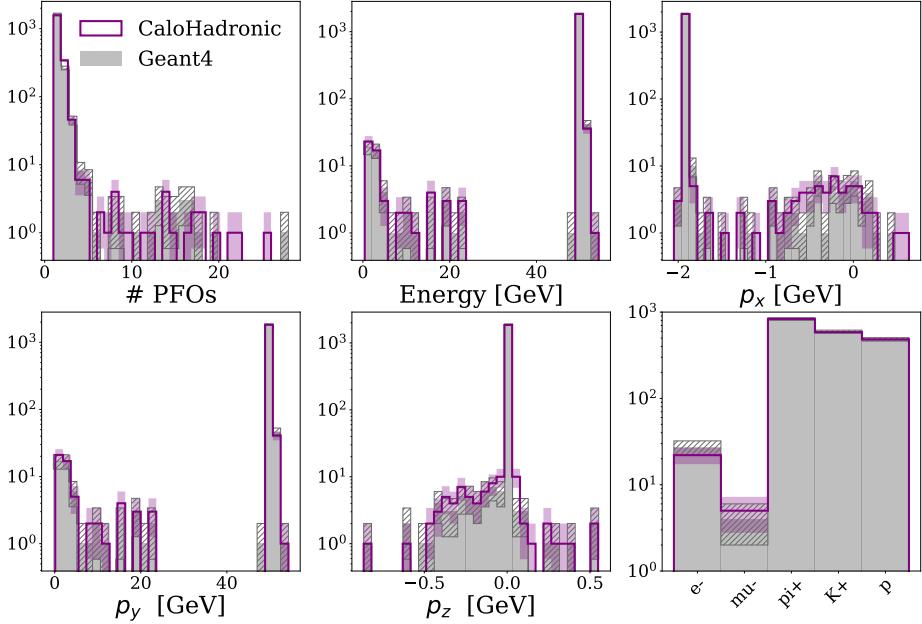
Figure 12 shows example event displays of the Pandora PFOs reconstructed for a Geant4 shower (left) and for a CALOHADRONIC shower (right). In total, 2000 such events are generated for both Geant4 and the model. After cuts were applied on the quality of the reconstructed track, a total of 1929 events for CALOHADRONIC and 1941 events for Geant4 remain.

### 5.2 Results

We now analyse the effects of reconstruction by studying the performance of the model compared to Geant4 in terms of key physics observables after reconstruction. The observables studied are the energy of the PFO, its momentum in each direction ( $p_x, p_y, p_z$ ) and the reconstructed particle type. Figure 13 shows distributions for each of these observables for all PFOs reconstructed across all events, as well as the number of PFOs (# PFOs) reconstructed in each event. The majority of PFOs are correctly reconstructed as  $\pi^+$ , however, a significant fraction ( 57%) are incorrectly reconstructed as other species of charged hadrons, namely  $K^+$  and  $p$ . A small fraction are incorrectly reconstructed as  $e^-$  and  $\mu^-$ . However, for all observables CALOHADRONIC provides an impressive

---

<sup>5</sup>This position is chosen to avoid the TPC cathode, which is positioned at  $z = 0$  in the global ILD coordinate system



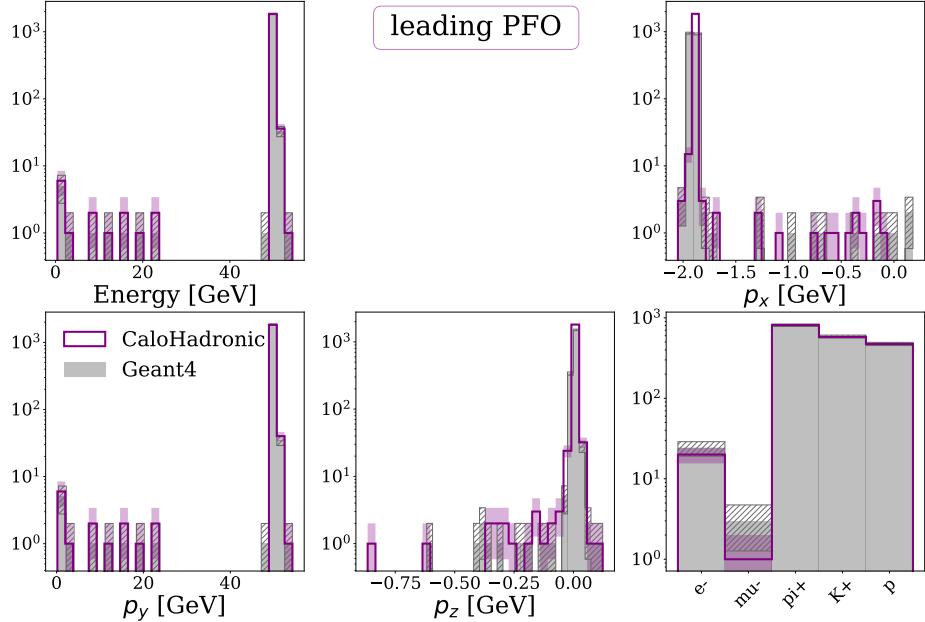
**Figure 13.** PFO-level observables after reconstruction of pions with PANDORAPFA for Geant4 (gray) and CALOHADRONIC (purple). Observables include distributions for the number of PFOs reconstructed per event (# PFOs, top left), PFO energy (top middle), PFO momentum in  $x$  ( $p_x$ , top left), PFO momentum in  $y$  ( $p_y$ , bottom left), PFO momentum in  $z$  ( $p_z$ , bottom middle), and the type of reconstructed particle (bottom right).

modeling of the various observables within uncertainties, including reproducing the deficiencies present in the reconstruction that are also present when simulating with Geant4.

Due to the complexity of hadronic showers, it is possible for multiple PFOs to be reconstructed from a single event. For this reason, we also show the reconstruction observables for the leading (i.e. highest energy) PFO per event in Figure 14. CALOHADRONIC again provides an impressive modeling of post-reconstruction PFO level observables.

## 6 Conclusion

To turn generative models from proof-of-concept studies into useful surrogates to be used in production, they need to be able to handle the various complexities of realistic calorimeters. This work demonstrates for the first time how a point-cloud based generative model can be extended to simultaneously simulate the ECal and HCal parts of a hadronic shower, emphasizing the flexibility of this approach. Both individual shower properties, as well as correlations (also across ECal and HCal components), are described well. A classifier test shows that — as for the generation of partial showers — the generative fidelity needs to be further improved. This is especially visible in the fine grained track-like structures observed in the HCal. However the degree of agreement between generative output and ground-truth simulation is of a similar or slightly better quality when also considering down-stream simulation.



**Figure 14.** PFO-level observables after reconstruction of pions with PANDORAPFA for Geant4 (gray) and CALOHADRONIC (purple). Only the leading PFO per event is shown. Observables include distributions for the leading PFO energy (top middle), leading PFO momentum in  $x$  ( $p_x$ , top left), leading PFO momentum in  $y$  ( $p_y$ , bottom left), leading PFO momentum in  $z$  ( $p_z$ , bottom middle), and the type of the leading reconstructed particle (bottom right).

Finally, the demonstrated strategy of sequentially conditioning several generative models might also be applied to other complex generative simulation tasks beyond calorimeters.

## 7 Acknowledgments

This research was supported in part by the Maxwell computational resources operated at Deutsches Elektronen-Synchrotron DESY, Hamburg, Germany. This project has received funding from the European Union’s Horizon 2020 Research and Innovation programme under Grant Agreement No 101004761. We acknowledge support by the Deutsche Forschungsgemeinschaft under Germany’s Excellence Strategy – EXC 2121 Quantum Universe – 390833306 and via the KISS consortium (05D23GU4, 13D22CH5) funded by the German Federal Ministry of Research, Technology and Space (BMFTR) in the ErUM-Data action plan. P.M. has benefited from support by the CERN Strategic R&D Programme on Technologies for Future Experiments [83].

## References

- [1] J. Apostolakis et al., *Detector Simulation Challenges for Future Accelerator Experiments*, *Front. in Phys.* **10** (2022) 913510.
- [2] S. Biedron et al., *Snowmass21 Accelerator Modeling Community White Paper*, in *Snowmass 2021*, 3, 2022 [[2203.08335](#)].
- [3] V.D. Elvira et al., *The Future of High Energy Physics Software and Computing*, in *Snowmass 2021*, 10, 2022 [[2210.05822](#)].
- [4] A. Adelmann et al., *New directions for surrogate models and differentiable programming for High Energy Physics detector simulation*, in *Snowmass 2021*, 3, 2022 [[2203.08806](#)].
- [5] S. Badger et al., *Machine learning and LHC event generation*, *SciPost Phys.* **14** (2023) 079 [[2203.07460](#)].
- [6] C. Krause, M. Faucci Giannelli, G. Kasieczka, B. Nachman, D. Salamani, D. Shih et al., *CaloChallenge 2022: A Community Challenge for Fast Calorimeter Simulation*, [2410.21611](#).
- [7] 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 [[1705.02355](#)].
- [8] 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 [[1712.10321](#)].
- [9] 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 [[1711.08813](#)].
- [10] M. Erdmann, L. Geiger, J. Glombitzka and D. Schmidt, *Generating and refining particle detector simulations using the Wasserstein distance in adversarial networks*, *Comput. Softw. Big Sci.* **2** (2018) 4 [[1802.03325](#)].
- [11] P. Musella and F. Pandolfi, *Fast and Accurate Simulation of Particle Detectors Using Generative Adversarial Networks*, *Comput. Softw. Big Sci.* **2** (2018) 8 [[1805.00850](#)].
- [12] M. Erdmann, J. Glombitzka and T. Quast, *Precise simulation of electromagnetic calorimeter showers using a Wasserstein Generative Adversarial Network*, *Comput. Softw. Big Sci.* **3** (2019) 4 [[1807.01954](#)].
- [13] D. Belayneh et al., *Calorimetry with deep learning: particle simulation and reconstruction for collider physics*, *Eur. Phys. J. C* **80** (2020) 688 [[1912.06794](#)].
- [14] A. Butter, S. Diefenbacher, G. Kasieczka, B. Nachman and T. Plehn, *GANplifying event samples*, *SciPost Phys.* **10** (2021) 139 [[2008.06545](#)].
- [15] ATLAS collaboration, *The Fast Simulation Chain in the ATLAS experiment*, *EPJ Web Conf.* **251** (2021) 03012.
- [16] S. Bieringer, A. Butter, S. Diefenbacher, E. Eren, F. Gaede, D. Hundhausen et al., *Calomplification — the power of generative calorimeter models*, *JINST* **17** (2022) P09028 [[2202.07352](#)].
- [17] B. Hashemi, N. Hartmann, S. Sharifzadeh, J. Kahn and T. Kuhr, *Ultra-high-granularity detector simulation with intra-event aware generative adversarial network and self-supervised relational reasoning*, *Nature Commun.* **15** (2024) 4916 [[2303.08046](#)].

- [18] ATLAS collaboration, *Deep Generative Models for Fast Photon Shower Simulation in ATLAS*, *Comput. Softw. Big Sci.* **8** (2024) 7 [[2210.06204](#)].
- [19] E. Buhmann, S. Diefenbacher, E. Eren, F. Gaede, G. Kasieczka, A. Korol et al., *Getting High: High Fidelity Simulation of High Granularity Calorimeters with High Speed*, *Comput. Softw. Big Sci.* **5** (2021) 13 [[2005.05334](#)].
- [20] E. Buhmann, S. Diefenbacher, E. Eren, F. Gaede, G. Kasieczka, A. Korol et al., *Decoding Photons: Physics in the Latent Space of a BIB-AE Generative Network*, *EPJ Web Conf.* **251** (2021) 03003 [[2102.12491](#)].
- [21] E. Buhmann, S. Diefenbacher, D. Hundhausen, G. Kasieczka, W. Korcari, E. Eren et al., *Hadrons, better, faster, stronger*, *Mach. Learn. Sci. Tech.* **3** (2022) 025014 [[2112.09709](#)].
- [22] J.C. Cresswell, B.L. Ross, G. Loaiza-Ganem, H. Reyes-Gonzalez, M. Letizia and A.L. Caterini, *CaloMan: Fast generation of calorimeter showers with density estimation on learned manifolds*, in *36th Conference on Neural Information Processing Systems: Workshop on Machine Learning and the Physical Sciences*, 11, 2022 [[2211.15380](#)].
- [23] S. Diefenbacher, E. Eren, F. Gaede, G. Kasieczka, A. Korol, K. Krüger et al., *New angles on fast calorimeter shower simulation*, *Mach. Learn. Sci. Tech.* **4** (2023) 035044 [[2303.18150](#)].
- [24] J. Sohl-Dickstein, E.A. Weiss, N. Maheswaranathan and S. Ganguli, *Deep Unsupervised Learning using Nonequilibrium Thermodynamics*, [1503.03585](#).
- [25] Y. Song and S. Ermon, *Generative modeling by estimating gradients of the data distribution*, in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, (Red Hook, NY, USA), Curran Associates Inc. (2019).
- [26] Y. Song and S. Ermon, *Improved techniques for training score-based generative models*, in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan and H. Lin, eds., vol. 33, pp. 12438–12448, Curran Associates, Inc., 2020,  
[https://proceedings.neurips.cc/paper\\_files/paper/2020/file/92c3b916311a5517d9290576e3ea37ad-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2020/file/92c3b916311a5517d9290576e3ea37ad-Paper.pdf).
- [27] J. Ho, A. Jain and P. Abbeel, *Denoising Diffusion Probabilistic Models*, [2006.11239](#).
- [28] Y. Song, J. Sohl-Dickstein, D.P. Kingma, A. Kumar, S. Ermon and B. Poole, *Score-Based Generative Modeling through Stochastic Differential Equations*, [2011.13456](#).
- [29] V. Mikuni and B. Nachman, *Score-based generative models for calorimeter shower simulation*, *Phys. Rev. D* **106** (2022) 092009 [[2206.11898](#)].
- [30] E. Buhmann, S. Diefenbacher, E. Eren, F. Gaede, G. Kasieczka, A. Korol et al., *CaloClouds: fast geometry-independent highly-granular calorimeter simulation*, *JINST* **18** (2023) P11025 [[2305.04847](#)].
- [31] F.T. Acosta, V. Mikuni, B. Nachman, M. Arratia, B. Karki, R. Milton et al., *Comparison of point cloud and image-based models for calorimeter fast simulation*, *JINST* **19** (2024) P05003 [[2307.04780](#)].
- [32] V. Mikuni and B. Nachman, *CaloScore v2: single-shot calorimeter shower simulation with diffusion models*, *JINST* **19** (2024) P02001 [[2308.03847](#)].
- [33] O. Amram and K. Pedro, *Denoising diffusion models with geometry adaptation for high fidelity calorimeter simulation*, *Phys. Rev. D* **108** (2023) 072014 [[2308.03876](#)].
- [34] C. Chen, O. Cerri, T.Q. Nguyen, J.R. Vlimant and M. Pierini, *Analysis-Specific Fast Simulation at the LHC with Deep Learning*, *Comput. Softw. Big Sci.* **5** (2021) 15.

- [35] C. Krause and D. Shih, *Fast and accurate simulations of calorimeter showers with normalizing flows*, *Phys. Rev. D* **107** (2023) 113003 [[2106.05285](#)].
- [36] C. Krause and D. Shih, *Accelerating accurate simulations of calorimeter showers with normalizing flows and probability density distillation*, *Phys. Rev. D* **107** (2023) 113004 [[2110.11377](#)].
- [37] S. Schnake, D. Krücker and K. Borras, *CaloPointFlow II Generating Calorimeter Showers as Point Clouds*, [2403.15782](#).
- [38] C. Krause, I. Pang and D. Shih, *CaloFlow for CaloChallenge dataset 1*, *SciPost Phys.* **16** (2024) 126 [[2210.14245](#)].
- [39] A. Xu, S. Han, X. Ju and H. Wang, *Generative machine learning for detector response modeling with a conditional normalizing flow*, *JINST* **19** (2024) P02003 [[2303.10148](#)].
- [40] M.R. Buckley, C. Krause, I. Pang and D. Shih, *Inductive simulation of calorimeter showers with normalizing flows*, *Phys. Rev. D* **109** (2024) 033006 [[2305.11934](#)].
- [41] F. Ernst, L. Favaro, C. Krause, T. Plehn and D. Shih, *Normalizing Flows for High-Dimensional Detector Simulations*, *SciPost Phys.* **18** (2025) 081 [[2312.09290](#)].
- [42] M. Omana Kuttan, K. Zhou, J. Steinheimer and H. Stöcker, *Towards a foundation model for heavy-ion collision experiments through point cloud diffusion*, [2412.10352](#).
- [43] L. Favaro, A. Ore, S.P. Schweitzer and T. Plehn, *CaloDREAM – Detector Response Emulation via Attentive flow Matching*, *SciPost Phys.* **18** (2025) 088 [[2405.09629](#)].
- [44] J. Brehmer, V. Bresó, P. de Haan, T. Plehn, H. Qu, J. Spinner et al., *A Lorentz-Equivariant Transformer for All of the LHC*, [2411.00446](#).
- [45] J. Birk, F. Gaede, A. Hallin, G. Kasieczka, M. Mozzanica and H. Rose, *OmniJet- $\alpha_C$ : Learning point cloud calorimeter simulations using generative transformers*, [2501.05534](#).
- [46] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez et al., *Attention is all you need*, *CoRR* **abs/1706.03762** (2017) [[1706.03762](#)].
- [47] W. Peebles and S. Xie, *Scalable diffusion models with transformers*, in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023 [[2212.09748](#)].
- [48] J. Leigh, C. Seitz, J.-H. Vogel, L. Funcke, L. Hartling, F. Keil et al., *PC-JeDi: Diffusion for Particle Cloud Generation in High Energy Physics*, *arXiv preprint* (2023) [[2303.05376](#)].
- [49] V. Mikuni, B. Nachman and G. Petree, *Fast point cloud generation with diffusion models in high energy physics*, *Phys. Rev. D* **108** (2023) 036025 [[2303.09593](#)].
- [50] A. Butter, G. Kasieczka, T. Plehn and T. Siewert, *Jet Diffusion versus JetGPT – Modern networks for the LHC*, *SciPost Phys. Core* **8** (2025) 026 [[2307.00018](#)].
- [51] V. Mikuni and B. Nachman, *Method to simultaneously facilitate all jet physics tasks*, *Phys. Rev. D* **111** (2025) 054015 [[2502.14652](#)].
- [52] M. Della Negra, P. Jenni and T.S. Virdee, *Journey in the search for the Higgs Boson: The ATLAS and CMS experiments at the Large Hadron Collider*, *Science* **338** (2012) 1560.
- [53] CMS collaboration, *Status and Plans for the CMS High Granularity Calorimeter Upgrade Project*, *J. Phys. Conf. Ser.* **2374** (2022) 012020 [[2111.12455](#)].
- [54] T. Collaboration, G. Aad, B. Abbott, J. Abdallah, A. Abdelalim, A. Abdelouahab et al., *Readiness of the atlas tile calorimeter for lhc collisions*, *European Physical Journal C* **70** (2010) 1193.

- [55] R.T.Q. Chen, Y. Rubanova, J. Bettencourt and D. Duvenaud, *Neural Ordinary Differential Equations*, [1806.07366](#).
- [56] T. Karras, M. Aittala, T. Aila and S. Laine, *Elucidating the design space of diffusion-based generative models*, [2206.00364](#).
- [57] ILD CONCEPT GROUP collaboration, *International Large Detector: Interim Design Report*, [2003.01116](#).
- [58] T. Behnke, J.E. Brau, B. Foster, J. Fuster, M. Harrison, J.M. Paterson et al., *The international linear collider technical design report - volume 1: Executive summary*, Tech. Rep. [arXiv:1306.6327](#), arXiv, <https://arxiv.org/abs/1306.6327> (2013).
- [59] 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.
- [60] The Geant4 Collaboration, *Geant4—a simulation toolkit*, *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **506** (2003) 250.
- [61] P. McKeown, E. Buhmann, T. Buss, S. Diefenbacher, E. Eren, F. Gaede et al., *Fast Simulation of Highly Granular Calorimeters with Generative Models: Towards a First Physics Application*, *PoS EPS-HEP2023* (2023) 568.
- [62] P. McKeown, *Development and Performance of a Fast Simulation Tool for Showers in High Granularity Calorimeters based on Deep Generative Models*, Ph.D. thesis, Hamburg U., Hamburg, 2024. 10.3204/PUBDB-2024-01825.
- [63] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan et al., *Pytorch: An imperative style, high-performance deep learning library*, in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, pp. 8024–8035, Curran Associates, Inc., 2019.
- [64] Y. Lipman, R.T.Q. Chen, H. Ben-Hamu, M. Nickel and M. Le, *Flow matching for generative modeling*, in *The Eleventh International Conference on Learning Representations*, 2023, <https://openreview.net/forum?id=PqvMRDCJT9t> [[2210.02747](#)].
- [65] K. Crowson, “k-diffusion.” <https://github.com/crowsonkb/k-diffusion>, 2022.
- [66] D.J. Rezende and S. Mohamed, *Variational inference with normalizing flows*, in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML’15, p. 1530–1538, JMLR.org, 2015.
- [67] G. Papamakarios, E. Nalisnick, D.J. Rezende, S. Mohamed and B. Lakshminarayanan, *Normalizing Flows for Probabilistic Modeling and Inference*, *J. Machine Learning Res.* **22** (2021) 2617 [[1912.02762](#)].
- [68] M.S. Albergo and E. Vanden-Eijnden, *Building Normalizing Flows with Stochastic Interpolants*, [2209.15571](#).
- [69] E. Buhmann, F. Gaede, G. Kasieczka, A. Korol, W. Korcari, K. Krüger et al., *CaloClouds II: ultra-fast geometry-independent highly-granular calorimeter simulation*, *JINST* **19** (2024) P04020 [[2309.05704](#)].
- [70] A.Q. Nichol and P. Dhariwal, *Improved denoising diffusion probabilistic models*, in *Proceedings of the 38th International Conference on Machine Learning*, M. Meila and T. Zhang, eds., vol. 139 of *Proceedings of Machine Learning Research*, pp. 8162–8171, PMLR, 18–24 Jul, 2021, <https://proceedings.mlr.press/v139/nichol21a.html>.

- [71] H. Sun, L. Yu, B. Dai, D. Schuurmans and H. Dai, *Score-based continuous-time discrete diffusion models*, in *The Eleventh International Conference on Learning Representations*, 2023, <https://openreview.net/forum?id=BYWWwSY2G5s>.
- [72] D.P. Kingma and R. Gao, *Understanding diffusion objectives as the ELBO with simple data augmentation*, in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023, <https://openreview.net/forum?id=NnMEadcdyD>.
- [73] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht et al., *On the spectral bias of neural networks*, in *Proceedings of the 36th International Conference on Machine Learning*, K. Chaudhuri and R. Salakhutdinov, eds., vol. 97 of *Proceedings of Machine Learning Research*, pp. 5301–5310, PMLR, 09–15 Jun, 2019, <https://proceedings.mlr.press/v97/rahaman19a.html>.
- [74] B. Mildenhall, P.P. Srinivasan, M. Tancik, J.T. Barron, R. Ramamoorthi and R. Ng, *Nerf: Representing scenes as neural radiance fields for view synthesis*, *CoRR* **abs/2003.08934** (2020) [[2003.08934](https://arxiv.org/abs/2003.08934)].
- [75] J. Ji, R. Zhao and M. Lei, *Latent diffusion transformer for point cloud generation*, *The Visual Computer* **40** (2024) 3903.
- [76] D.P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, 12, 2014 [[1412.6980](https://arxiv.org/abs/1412.6980)].
- [77] Y. Zhang, C. Chen, Z. Li, T. Ding, C. Wu, D.P. Kingma et al., *Adam-mini: Use fewer learning rates to gain more*, in *The Thirteenth International Conference on Learning Representations*, 2025, <https://openreview.net/forum?id=iBExhaU3Lc>.
- [78] R. Rombach, A. Blattmann, D. Lorenz, P. Esser and B. Ommer, *High-resolution image synthesis with latent diffusion models*, [2112.10752](https://arxiv.org/abs/2112.10752).
- [79] L.N. Smith, *Cyclical learning rates for training neural networks*, in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 464–472, 2017, [DOI](https://doi.org/10.1109/WACV.2017.8195000).
- [80] L.N. Smith and N. Topin, *Super-convergence: Very fast training of neural networks using large learning rates*, [1708.07120](https://arxiv.org/abs/1708.07120).
- [81] M.A. Thomson, *Particle Flow Calorimetry and the PandoraPFA Algorithm*, *Nucl. Instrum. Meth. A* **611** (2009) 25 [[0907.3577](https://arxiv.org/abs/0907.3577)].
- [82] J.S. Marshall and M.A. Thomson, *The Pandora Software Development Kit for Pattern Recognition*, *Eur. Phys. J. C* **75** (2015) 439 [[1506.05348](https://arxiv.org/abs/1506.05348)].
- [83] C. Joram et al., *Extension of the R&D Programme on Technologies for Future Experiments*, Tech. Rep. CERN-EP-RDET-2023-001, CERN, <https://cds.cern.ch/record/2850809> (2023).
- [84] J. Ji, R. Zhao and M. Lei, *Latent diffusion transformer for point cloud generation*, *The Visual Computer* **40** (2024) 3903.
- [85] A.L. Maas, *Rectifier nonlinearities improve neural network acoustic models*, 2013, <https://api.semanticscholar.org/CorpusID:16489696>.
- [86] T. Buss, F. Gaede, G. Kasieczka, C. Krause and D. Shih, *Convolutional L2LFlows: generating accurate showers in highly granular calorimeters using convolutional normalizing flows*, *JINST* **19** (2024) P09003 [[2405.20407](https://arxiv.org/abs/2405.20407)].
- [87] C. Krause and D. Shih, *Fast and accurate simulations of calorimeter showers with normalizing flows*, *Phys. Rev. D* **107** (2023) 113003.
- [88] D. Lopez-Paz and M. Oquab, *Revisiting classifier two-sample tests*, in *International Conference on Learning Representations*, 2017, <https://openreview.net/forum?id=SJkXfE5xx>.

- [89] C.R. Rao, *Linear Statistical Inference and Its Applications*, Wiley Series in Probability and Statistics, John Wiley & Sons, Inc., second ed. (1973), [10.1002/9780470316436](https://doi.org/10.1002/9780470316436).

## A Model details and hyper-parameters

In tables 2 and 3 a detailed view of the hyper-parameters used in the trainings of POINTCOUNTFM ECAL EDM-DIFFUSION and HCAL EDM-DIFFUSION is presented.

Category	Hyperparameter	ECAL EDM-DIFFUSION	HCAL EDM-DIFFUSION
GENERAL	<b>only_HCal</b>	True	False
	<b>only_ECal</b>	False	True
	<b>Seed</b>	45	455
	<b>Training Hardware</b>	1 x NVIDIA® A100	4 x NVIDIA® A100
	<b>Optimizer</b>	AdamMini	AdamMini
	<b>Loss Type</b>	EDM-Monotonic	EDM-Monotonic
EDM FEATURES	<b>KL Weight</b>	$10^{-3}$	$10^{-3}$
	<b>KLD Min</b>	1.0	1.0
	<b>Schedule Mode</b>	Quadratic	Quadratic
	<b>EMA Type</b>	Inverse, Power=0.6667, Max=0.9999	Inverse, Power=0.6667, Max=0.9999
TRANSFORMER	<b>Transformer Decoder Layers</b>	0	2
	<b>Transformer Encoder Layers</b>	3	4
	<b>Embedding Dimension</b>	128	128
	<b>Number of heads</b>	8	16
	<b>Feedforward dim</b>	512	512
	<b>Dropout Rate</b>	0.1	0.1
FUORIER LAYER	<b>Include Input</b>	True	True
	<b>Max Frequency</b>	16	16
	<b>Number of Frequencies</b>	32	32
	<b>Log Sampling</b>	True	True
	<b>periodic Functions</b>	[sin, cos]	[sin, cos]
DATA	<b>Granularity</b>	x9	x9
	<b>Features</b>	4	4
	<b>Ecal Features</b>	-	4
	<b>Cond. Features</b>	30 + 1	48 + 1
	<b>Cond. Normalization</b>	True	True
	<b>Log energy</b>	True	True
	<b>Batch Size</b>	32	32
	<b>Dataloader Workers</b>	40	40
SCHEDULER	<b>LR Scheduler</b>	OneCycleLR	OneCycleLR
	<b>LR Start / Max / End</b>	$3 \cdot 10^{-5} / 3 \cdot 10^{-4} / 1 \cdot 10^{-6}$	$3 \cdot 10^{-5} / 3 \cdot 10^{-4} / 1 \cdot 10^{-6}$
	<b>Warmup Steps</b>	300k	500k
	<b>Total number of gradient steps</b>	2M	3M
SAMPLING	<b>Sigma Data</b>	0.5	0.5
	<b>Sigma Sampling</b>	Lognormal( $\mu=0, \sigma=1$ )	Lognormal( $\mu=0, \sigma=1$ )
	<b>ODE Solver</b>	Heun	Heun
	<b># Sampling Steps</b>	30	30
	<b>Sigma Min / Max</b>	0.01 / 10.0	0.01 / 10.0
	<b>Rho / s_churn / s_noise</b>	7.0 / 0.0 / 1.0	7.0 / 0.0 / 1.0

**Table 2.** Summary of ECAL EDM-DIFFUSION and HCAL EDM-DIFFUSION Training Configuration

## B Compression block

Taking inspiration from [84], a compression block was implemented. The compression block is a simple neural network block designed to reduce a variable-sized set of point features into a fixed number of representative “tokens”, 10 in this case. The process happens in two main steps:

Category	Hyperparameter	POINTCOUNTFM
DATA	<b>Train/Val Split</b>	90% / 10%
	<b>Pin Memory</b>	True
	<b>Workers</b>	20
	<b>Shuffle</b>	True
ARCHITECTURE	<b>Num Inputs</b>	78
	<b>Conditioning Inputs</b>	1
	<b>Time Embedding Dim</b>	6
	<b>Hidden Dims</b>	[128, 256, 512, 256, 128]
TRAINING	<b>Device</b>	NVIDIA® A100
	<b>Optimizer</b>	Adam-mini
	<b>Scheduler</b>	OneCycleLR
	<b>Max Learning Rate</b>	0.001
	<b>Warm up epochs</b>	900
GENERATION	<b>Batch Size</b>	3000
	<b>Epochs</b>	3000
	<b>ODE Solver</b>	Heun
	<b># Sampling Steps</b>	200

**Table 3.** Configuration summary for the POINTCOUNTFM.

- First, the model looks at each input point and decides how much it should contribute to each token. It does this firstly by applying a linear layer to end up with the token’s dimensionality and then using a softmax function. The outputs are per-point weights that tell the model how important each point is for each token with a probability value.
- A linear layer is applied to the input to map it to the output dimensionality. In this case the output dimension is the same as the input one, i.e. the number of features (4).
- After that, the compression block maps the points to the tokens. It does this by computing a weighted average of each point using the weights from step 1 (how important each point was for that token).

The result is a fixed-size output (one vector per token) that summarizes the input set.

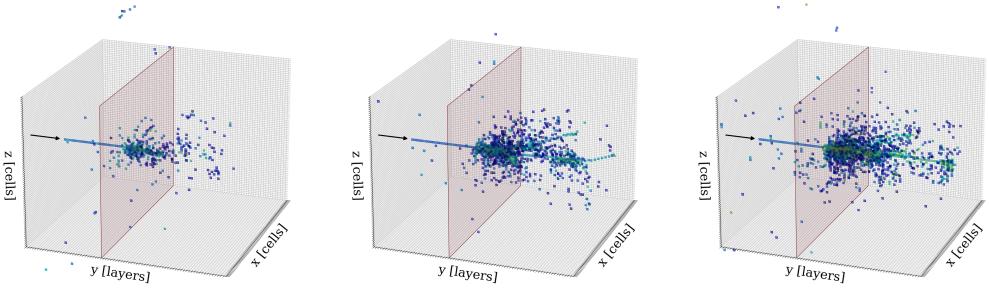
## C 3D $\pi^+$ showers

In this section, we present additional 3D plots of  $\pi^+$  showers to illustrate the state of the shower-structure simulation. Figure 15 shows examples of showers simulated by Geant4 with incident energies of 15 GeV, 50 GeV and 85 GeV.

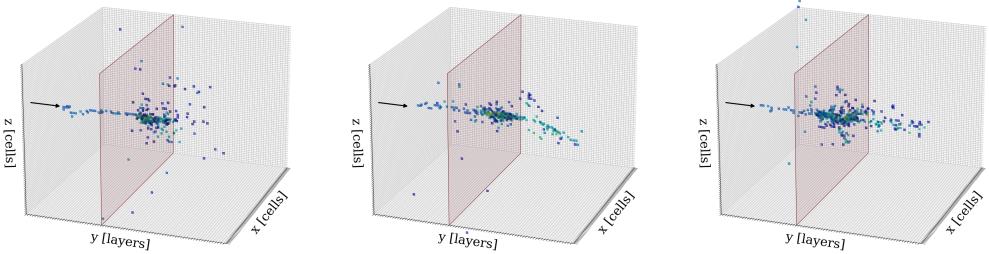
Figures 16, 17, and 18 show three examples of 15 GeV, 50 GeV and 85 GeV showers generated by CALOHADRONIC.

## D POINTCOUNTFM histograms

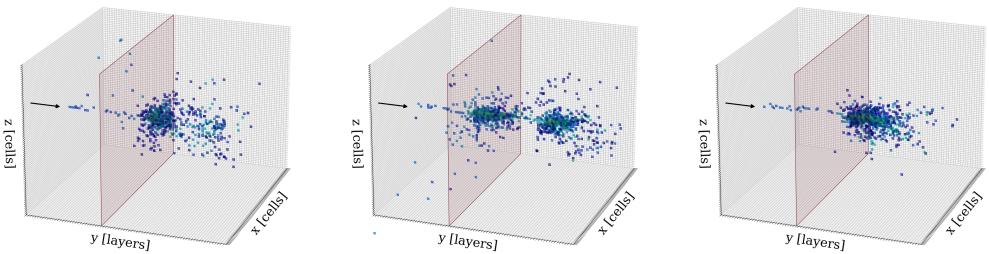
Figure 19 shows the generation of the points per layer in both ECal and HCal performed by the POINTCOUNTFM. One can see that the majority of points is usually around layer 30 (beginning of



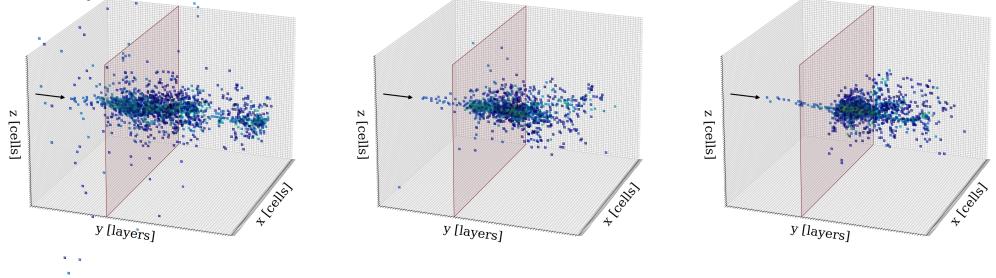
**Figure 15.** 3D view of a 15 GeV (left), a 50 GeV (center), and a 85 GeV (right) Geant4 shower. The color represents the energy deposition in the cells. The red plane represents the division between ECal and HCal.



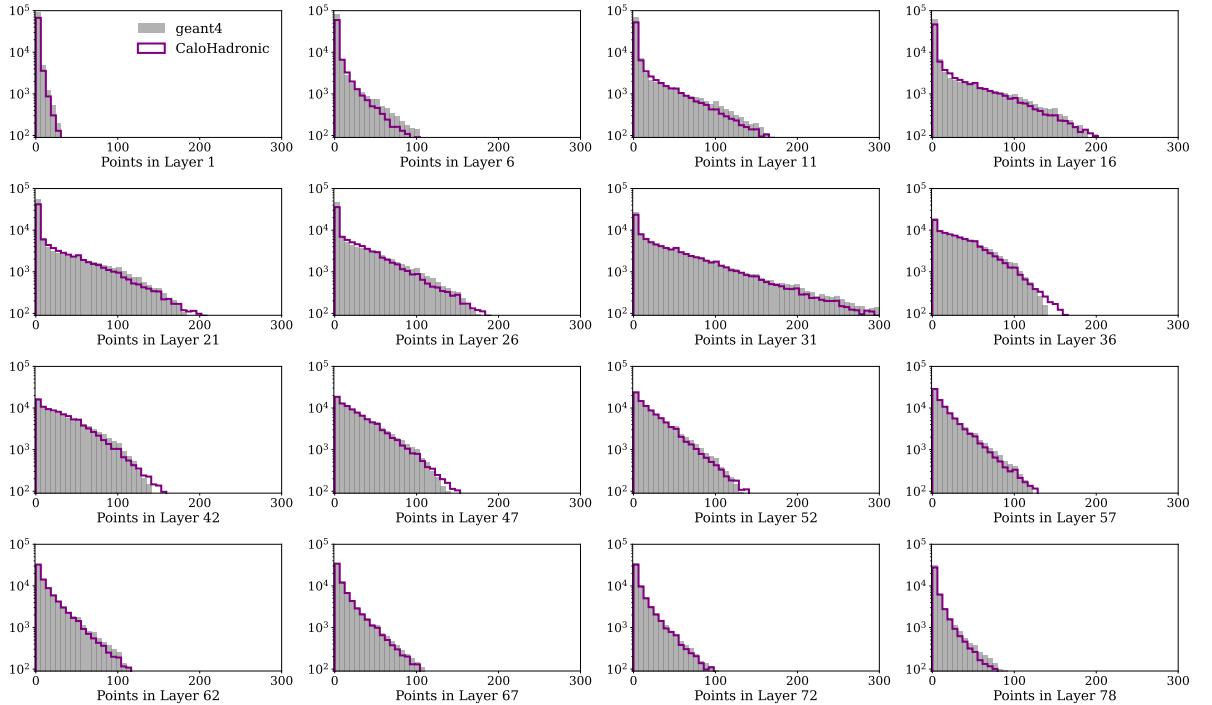
**Figure 16.** 3D view of three 15 GeV CALO HADRONIC showers. The color represents the energy deposition in the cells. The red plane represents the division between ECal and HCal.



**Figure 17.** 3D view of three 50 GeV CALO HADRONIC showers. The color represents the energy deposition in the cells. The red plane represents the division between ECal and HCal.



**Figure 18.** 3D view of three 85 GeV CALO HADRONIC showers. The color represents the energy deposition in the cells. The red plane represents the division between ECal and HCal.



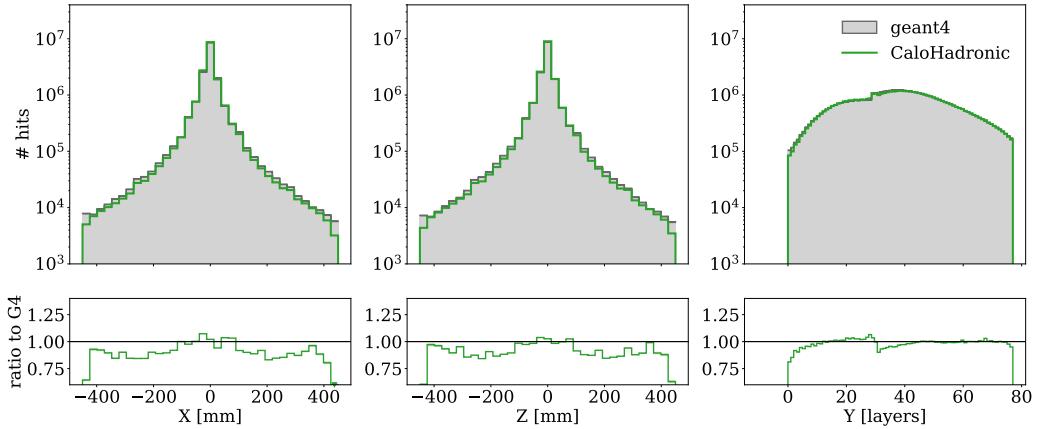
**Figure 19.** Histograms of the number of points per layer for Geant4 (gray) and POINTCOUNTFM (purple). All distributions are calculated for 100 000 showers with a uniform distribution of incident particle energies between 10 and 90 GeV.

the HCal), which is where a shower most likely starts, while in the first and last layers the number of points decreases.

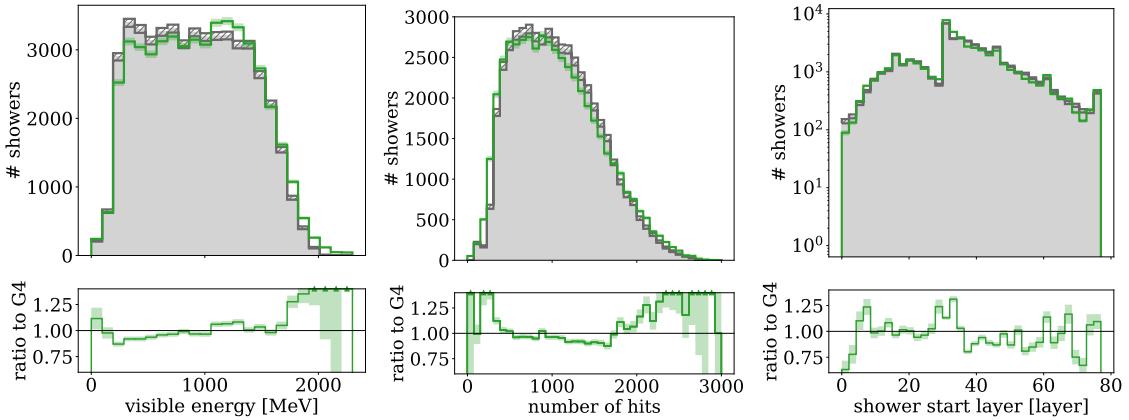
A high quality modeling of these distributions is required as the points per layer output is used as conditioning to generate the hadronic showers and to further calibrate the number of points per layer in the shower. POINTCOUNTFM effectively ensures the required precision and accuracy for this process.

## E CALOHADRONIC further histograms

Figures 20 and 21 show additional CALOHADRONIC distributions for completeness.



**Figure 20.** Distribution of hit  $x$  (left),  $y$  (middle), and  $z$  (right) positions calculated from 50000 showers. All distributions are calculated with a uniform distribution of incident particle energies between 10 and 90 GeV. The error band corresponds to the statistical uncertainty in each bin.



**Figure 21.** Histograms of the total visible energy (left), total number of hits (center), and shower start layer (right) for Geant4 and CALOHADRONIC. All distributions are calculated with 50000 events sampled from a uniform distribution of incident particle energies between 10 and 90 GeV. The bottom panel provides the ratio to Geant4. The error band corresponds to the statistical uncertainty in each bin.

## F Evaluation Scores

We investigate the performance of CALOHADRONIC by calculating scores from high level calorimeter shower observables. The evaluation in this work relies on two metrics: the Wasserstein distance, which measures distributional similarity between features, and the quantile KL divergence, which compares the divergence between quantile distributions.

This provides a quantitative measure of the fidelity observed in the plots presented in Sec. 4.1, allowing for an objective assessment rather than relying solely on a visual comparison of distributions. The evaluation scores are computed with 5 batches of 10,000 showers.

The following observables are considered in order to calculate the one-dimensional scores: the number of hits (cells with energy depositions above 0.02 MeV threshold)  $N_{\text{hits}}$ , the cell energy  $E_{\text{cell}}$ , the center of gravity in the  $x$ -,  $y$ -, and  $z$ -directions  $cog_x$ ,  $cog_y$  and  $cog_z$ , energy sum  $E_{\text{sum}}$ , the longitudinal energy  $E_{\text{long}}$ , and the radial energy  $E_{\text{radial}}$ .

$(x10^{-2})$	$cog_x$	$cog_y$	$cog_z$	$Y_{\text{start}}$	$E_{\text{cell}}$	$E_{\text{sum}}$	$E_{\text{radial}}$	$E_{\text{long}}$	$N_{\text{hits}}$
normalized WD	$8.4 \pm 0.8$	$3.8 \pm 0.4$	$8.0 \pm 0.7$	$5.2 \pm 0.2$	$2.8 \pm 0.4$	$9 \pm 1$	$0.05 \pm 0.02$	$76 \pm 37$	$2.8 \pm 0.4$
quantile KL	$1.1 \pm 0.3$	$0.6 \pm 0.1$	$0.9 \pm 0.2$	$2.1 \pm 0.2$	$0.3 \pm 0.1$	$0.68 \pm 0.07$	$0.2 \pm 0.2$	$0.6 \pm 0.7$	$0.3 \pm 0.1$

**Table 4.** Wasserstein distances for several physics observables between generated and test data. The Wasserstein distances and the quantile KL are numerically evaluated using five batches of 10,000 showers. Shown are the mean and the standard deviation over the five resulting values.

### F.1 Classifier Scores

We further compare the showers generated by the model to the Geant4 simulation by training a fully connected high-level classifier to distinguish between model generated and Geant4 simulated showers. The 5 input shower observables are the three center of gravity variables, the number of hits and energy ratio (total visible energy divided by the incident energy). 150k Geant4 showers and 500k showers generated by CALOHADRONIC were used. A standard scaling of the inputs is applied. A 60%, 20%, 20% data split is applied for training set, validation and test set, respectively. The classifier is implemented as a fully connected neural network with three layers (containing 8, 8, 1 nodes respectively) with LeakyReLU [85] activation functions. It is trained with the Adam optimizer [76] for 100 epochs using a binary cross-entropy loss. The final model epoch is chosen based on the lowest validation loss.

To evaluate the performance of the classifier, we use two metrics: the area under the receiver operating characteristic curve (AUC) and the Jensen–Shannon divergence (JSD), both computed on the test set. These metrics are commonly used in the evaluation of generative models in high-energy physics, as demonstrated in previous works such as Ref. [30, 36, 40, 86–88]. The AUC reflects the classifier’s ability to distinguish between Geant4 and model-generated data. A perfect separation yields an AUC of 1.0, while an AUC of 0.5 indicates total confusion—implying that the generated data is indistinguishable from the simulated ones.

The JSD, on the other hand, is a symmetric and bounded measure of the difference between two probability distributions, often used to quantify how similar two datasets are. In this context, it measures the divergence between the classifier’s predicted probability distributions for simulated

and generated samples. A JSD of 0 indicates perfect overlap (i.e., the distributions are identical), while higher values suggest greater divergence between the real and generated data.

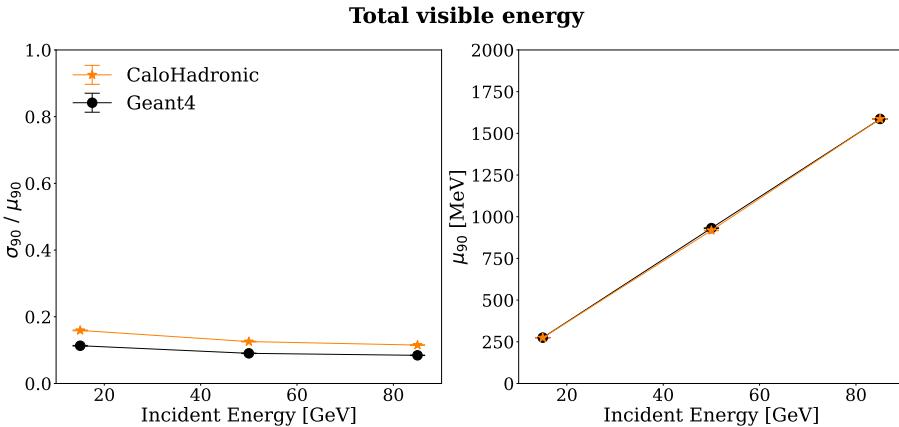
To ensure robustness, the classifier was trained five times using different splits of the data into training, validation, and test sets. In Tab. 5 we present the mean AUC and standard deviation of these five classifier trainings for CALOHADRONIC.

Simulator	AUC	JSD
CALOHADRONIC	$0.79 \pm 0.03$	$0.2354 \pm 0.0005$

**Table 5.** Results of the high level classifier test. Model performance comparison with area under the receiver operating characteristic curve (AUC) score and Jensen–Shannon divergence (JSD). Shown are the mean and standard deviation over five random network initializations.

## G Resolution and Linearity

Figure 22 shows the relative resolution, computed by dividing the standard deviation by the mean ( $\sigma / \mu$ ) versus the incident energy, and the linearity, hence the mean ( $\mu$ ) versus the incident energy, of the total visible energy per shower. Incident energies of 15 GeV, 50 GeV, and 85 GeV are shown. Geant4 showers are sampled within  $\pm 1$  MeV of the target energy; CALOHADRONIC showers are newly generated at the same incident energies. The mean and root-mean-square of the 90% core of these distributions, labeled  $\mu_{90}$  and  $\sigma_{90}$ , is calculated for all energies. It should be noted that the resolution has not been corrected for the absorber thickness.



**Figure 22.** Relative resolution  $\sigma_{90} / \mu_{90}$  (on the left) and linearity  $\mu_{90}$  (on the right) of the 90% core of the total visible energy per shower for Geant4 and CALOHADRONIC. The points are taken at incident energies of 15 GeV, 50 GeV, and 85 GeV. Geant4 showers are sampled within  $\pm 1$  MeV of the target energy; CALOHADRONIC showers are newly generated at the same incident energies.

In figure 22, in the linearity plots, the error on the mean is simply  $\sigma / \sqrt{N}$ . The relative resolution uncertainty was computed following [89]. Using the delta method approximation, the

resulting formula is:

$$\text{se}(s) \approx \frac{1}{2s} \sqrt{\frac{1}{n} \left( \mu_4 - \frac{n-3}{n-1} \sigma^4 \right)} \quad (\text{G.1})$$

where  $n$  is the sample size,  $\mu_4 = E[(X - \mu)^4]$ ,  $\sigma^4$  is the square of the population variance and  $s$  is the sample standard deviation.

Figure 22 shows that the resolution of CALOHADRONIC is slightly worse than Geant4, while the linearity plot shows good agreement between the two. Fitting a function of the form  $a+b/\sqrt{E_0}+c/E_0$  to the incident energy ( $E_0$ ) versus relative resolution ( $\sigma_{90}/\mu_{90}$ ) distribution yields the following parameters for Geant4 (CaloHadronic): a:  $0.063 \pm 0.003$  ( $0.069 \pm 0.005$ ), b:  $0.16 \pm 0.04$  ( $0.40 \pm 0.06$ ), c:  $0.16 \pm 0.04$  ( $-0.2 \pm 0.1$ ). However, it should be noted that these values cannot be directly interpreted as resolution components, as they average over different materials and thicknesses.